

PENILAIAN UJIAN BERTIPE ESSAY MENGGUNAKAN METODE TEXT SIMILARITY

Meiyanto Eko Sulisty¹, Ristu Saptono², Adam Asshidiq³

Jurusan Informatika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Sebelas Maret, Surakarta

¹mekosulistyo@staff.uns.ac.id, ²ristu.saptono@staff.uns.ac.id, ³a.asshidiq@gmail.com

Abstrak

Exam is one way to measure the performance of students in the learning process. The type of test that is often tested on every student's learning activity is the description of the type of exam (essay). This study focuses on the manufacture of automatic assessment to test the extension of type descriptions using text similarity. This method has several stages such as text mining, text preprocessing algorithm to process stemming Nazief-Adriani, transforming text, then the assessment using tf-idf method, as well as cosine similarity. Results of this study is the final value assessment system using text similarity did not differ significantly from the results of expert assessment. This is proven by testing using paired T test with degrees of freedom (α) of 5%, where the value $t_{tabel} > t_{hitung}$ ($2,045 > 0.3755$) as well as the significant value $>$ degrees of freedom ($0.71 > 0.05$). However, there are several different types of questions that a significant result because there are unique characters that terrekam in the database as well as answers to the key does not contain relevant keywords that match the correct answer.

Keywords: *essay exams, automatic scoring, text similarity, keyword*

Abstract

Ujian adalah salah satu cara untuk mengukur performa siswa dalam proses belajar. Jenis ujian yang sering diujikan pada setiap kegiatan belajar siswa adalah tipe ujian uraian (essay). Penelitian ini berfokus pada pembuatan ekstensi penilaian otomatis untuk ujian bertipe uraian dengan menggunakan metode *text similarity*. Metode ini mempunyai beberapa tahapan seperti *text mining*, *text preprocessing* dengan proses *stemming* menggunakan algoritma Nazief-Adriani, *text transforming*, lalu penilaiannya menggunakan metode *tf-idf*, serta *cosine similarity*. Hasil penelitian ini adalah nilai akhir hasil penilaian menggunakan sistem *text similarity* tidak berbeda signifikan dengan hasil penilaian dari pakar. Hal ini dibuktikan dengan pengujian menggunakan uji T berpasangan dengan derajat kebebasan (α) sebesar 5%, dimana nilai $t_{tabel} > t_{hitung}$ ($2.045 > 0.3755$) serta nilai signifikansi $>$ derajat kebebasan ($0.71 > 0.05$). Namun, terdapat beberapa tipe soal yang hasilnya berbeda signifikan dikarenakan terdapat karakter unik yang terrekam di dalam *database* serta jawaban pada kunci tidak mengandung *keyword* yang sesuai dengan relevansi jawaban yang benar.

Kata Kunci : *ujian essay, penilaian otomatis, text similarity, keyword*

1. PENDAHULUAN

Penilaian sering digunakan untuk mengukur performa peserta didik^[1]. Penilaian dianggap sebagai sebagai salah satu komponen utama dalam pembelajaran di kelas untuk membantu siswa dalam proses belajar^[2]. Salah satu cara penilaian dalam pembelajaran di kelas adalah dengan ujian. Jenis ujian yang biasa digunakan dalam setiap proses belajar mengajar adalah tipe pilihan ganda (*multiple choice*), benar atau salah (*true or false*), serta uraian (essay). Di dunia teknologi informasi sekarang ini, telah dikembangkan cara ujian yang praktis yang dapat diakses menggunakan komputer / laptop, yaitu dengan ujian *online*. Penggunaan web sebagai sarana untuk mengerjakan ujian dapat menjadi suatu media teknologi revolusi ujian. Namun, yang banyak digunakan dan dikembangkan pada saat ini hanya ujian bertipe *multiple choice* serta *true or false*. Sampai saat ini belum ada yang membuat ujian online bertipe uraian (essay), serta belum banyak metode yang digunakan untuk penilaian soal bertipe uraian. Soal yang digunakan merupakan soal yang jawabannya menggunakan teks dan tidak menggunakan

simbol matematis, salah satunya mata pelajaran Geografi. Metode yang digunakan untuk pembuatan sistem penilaian otomatis adalah metode *text similarity* karena dapat digunakan pada soal yang jawabannya lebih banyak berupa teks dan bukan symbol.

2. PENELITIAN TERKAIT

Beberapa penelitian yang berkaitan dengan ujian essay dan *text mining* diantaranya oleh Arief Hidayat dan Aris Sugiarto yang membahas sistem kuis online adaptif yang dapat dimanfaatkan sebagai ekstensi *content management system*. Sistem kuis adaptif ini menjadi lebih personal karena model pertanyaan yang disajikan secara khusus dirancang bagi mahasiswa sesuai dengan tingkat kemampuan mereka. Mahasiswa akan lebih mengenal kekuatan dan kelemahan dalam proses belajar mereka karena mereka tidak akan menuju ke tingkat kesulitan yang lebih tinggi jika mereka tidak memenuhi nilai yang dipersyaratkan pada tingkat tertentu (Hidayat, Arief & Sugiharto, Aris, 2012).

Sapda Rifa Agri Negara dalam penelitian yang berjudul "Pembuatan Sistem Ujian Kompetensi Pelatihan Online". Penelitian dilakukan dengan mengamati sistem secara langsung sehingga didapatkan hasil analisis mengenai kebutuhan sistem yang akan dibuat. Studi pustaka juga dilakukan untuk mencari sumber-sumber lain yang dapat menjadi dasar dan acuan dalam membuat sistem ujian *online*. Sistem ujian *online* dirancang dengan menggunakan alat bantu perancangan sistem. Dapat disimpulkan bahwa sistem ujian *online* telah didesain dan diimplementasikan dengan fasilitas meliputi manajemen paket soal ujian dan manajemen peserta ujian yang nantinya akan menghasilkan *output* berupa nilai hasil ujian (Negara, Sabda R.A. , 2011)

Riky Bagus Muhajir meneliti dengan judul "Metode *Similarity-Mashup* untuk *Framework* Modul *Relevant Content* pada *Content Management System* (CMS)". Penelitian mengenai *relevant content* dari berbagai informasi yang diperoleh dari berbagai website. *Relevant content* didapat dengan menghitung kemiripan meta data antara kandidat *relevant content* dengan konten utama. Hasil dari penelitian ini didapatkan bahwa metode *similarity-mashup* untuk modul *relevant content* telah dapat terintegrasi pada CMS dengan baik. Selain itu, dapat dilihat bahwa seringkali berbagai kandidat *relevant content* yang berasal dari luar sistem (*mashup*) memiliki nilai *similarity* yang lebih tinggi dibandingkan dengan kandidat yang berasal dari luar system (Muhajir, Riky Bagus. , 2012).

Lalu, penelitian yang dilakukan oleh Bambang Kurniawan yang berjudul "Klasifikasi Konten Berita dengan Metode *Text Mining*". Penelitian mengenai pengklasifikasian berita yang ada di dalam portal berita secara otomatis. Metode yang digunakan dalam penelitian ini adalah metode *Text mining*, yaitu metode klasifikasi yang merupakan variasi dari data mining berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar. Sedangkan algoritma yang digunakan adalah algoritma *naïve bayes classifier*, yaitu algoritma pendukung untuk melakukan klasifikasi. Hasil dari penelitian ini menghasilkan sistem klasifikasi berita berbasis web dengan menggunakan bahasa pemrograman PHP dan database MySQL menunjukkan bahwa berita testing bisa terklasifikasi secara otomatis seluruhnya (Kurniawan, Bambang., Effendi, Syahril., Sitompul, Opim Salim, 2012).

Dan yang terakhir, riset yang dilakukan oleh Ledi Agusta mengenai "Perbandingan Algoritma *Stemming Porter* dengan Algoritma Nazief & Adriani untuk *Stemming* Dokumen Teks Bahasa Indonesia". Riset ini bertujuan untuk membandingkan kemampuan dan ketepatan algoritma untuk melakukan *stemming* dokumen teks Bahasa Indonesia. Proses *stemming* pada teks berbahasa Indonesia lebih rumit/kompleks karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan *root word* dari sebuah kata. Algoritma *stemming* yang dibandingkan yaitu algoritma Porter dan algoritma Nazief & Adriani untuk proses *stemming* pada teks ber-Bahasa Indonesia. Efektifitas algoritma *stemming* diukur berdasarkan beberapa parameter, seperti kecepatan proses, keakuratan, dan kesalahan, sehingga diketahui algoritma manakah yang lebih cepat, lebih akurat atau yang lebih banyak melakukan kesalahan *stemming*. Hasil dari penelitian ini adalah bahwa algoritma Nazief & Adriani lebih cepat dan akurat dalam proses *stemming* teks ber-Bahasa Indonesia dibandingkan dengan algoritma Porter (Agusta, Ledi, 2009).

3. LANDASAN TEORI

3.1. Kuis Online

Kuis adalah sekumpulan pertanyaan yang digunakan untuk mengukur pengetahuan atau keterampilan siswa untuk tujuan memberikan umpan balik untuk menginformasikan kepada siswa yang bersangkutan (QuestionMark & League, 2004). Kuis dapat digunakan untuk menghasilkan beberapa jenis pertanyaan: pertanyaan pilihan ganda, mengisi kekosongan, respon numerik, pencocokan daftar, jawaban singkat, respon hot spot dan benar / salah untuk berbagai tujuan penilaian seperti sebagai penilaian formatif dan sumatif (Quinn, D.; & Reid, I. 2003) :

1. Penilaian Formatif biasanya disebut *self-assessment*, yang memungkinkan peserta didik untuk belajar sendiri, tanpa bimbingan guru (QuestionMark & League, 2004).
2. Penilaian Sumatif memberikan penilaian kinerja secara keseluruhan dengan memberikan grading kuantitatif dan kualitatif dari prestasi belajar peserta didik, yang sering dikenal sebagai "ujian jangka menengah" atau "ujian akhir" (QuestionMark & League, 2004).

Namun, kuis biasanya mengacu pada penilaian formatif^[8]. Sebelum adanya komputer dan *internet*, kuis ini dilakukan secara manual menggunakan kertas dan pensil. Guru dibutuhkan untuk mempersiapkan pertanyaan-pertanyaan dan kemudian mendistribusikannya secara manual untuk peserta didik mereka. Seiring pertumbuhan komputer dan teknologi, kuis dilakukan melalui Internet. Contoh komersial kuis online yang tersedia di pasaran saat ini QuestionMark (<http://www.questionmark.com>) dan Can Studios (<http://www.the-can.com>).

3.2. Text Mining

Sebelum melakukan proses *similarity* antar dokumen, perlu dilakukan pengolahan terlebih dahulu. *Text mining* digunakan untuk menggali informasi pada dokumen yang tidak terstruktur (Dorre, J, Gerstl, P & Seiffert, R., 1999). Sebelum melakukan *text mining*, terlebih dahulu melakukan proses perubahan bentuk data tidak menjadi nilai-nilai numerik. Setelah data diubah ke dalam nilai numerik, maka data tersebut dapat diolah lebih lanjut. Proses *text mining* dibagi menjadi ke dalam tiga buah proses, yaitu *Text Preprocessing*, *Text Transforming*, dan *Pattern Discovery*.

3.2.1. Text Preprocessing

Tahapan awal dalam *text mining* adalah *text preprocessing*. Tahap ini bertujuan untuk mempersiapkan teks yang menjadi sumber data untuk diproses ke tahap selanjutnya. Terdapat beberapa hal yang dilakukan dalam tahap ini, yaitu tindakan yang bersifat kompleks seperti *part-of-speech* (pos), *tagging*, *parse tree*, dan tindakan sederhana seperti *tokenization*. *Tokenization* merupakan proses pengolahan *token* yang terdapat dalam rangkaian teks (Grossmann, D & Frieder, O., 2004). Dalam pembuatan indeks istilah dokumen dipecah menjadi unit-unit lebih kecil, misalnya berupa kata, frasa, atau kalimat. Unit pemrosesan tersebut disebut *token*. Tahapan ini juga menghilangkan karakter-karakter tertentu, seperti tanda baca, *html tag*, dan mengubah semua *token* huruf kecil (*case folding*).

3.2.2. Text Transforming

Pada tahap ini dilakukan proses penyaringan (*filtration*) dengan menghilangkan *stop-word*. *Stop-word* merupakan kata yang sering muncul dalam dokumen tetapi tidak memiliki makna yang berarti. *Filtering* berfungsi untuk mengurangi daftar kata indeks yang mempercepat *indexing*^[11]. Penghapusan *stop-word* dari dalam suatu koleksi dokumen pada suatu waktu membutuhkan banyak waktu. Solusinya adalah dengan menyusun suatu pustaka *stop-word* atau *stop list* dari *term* yang akan dihapus. Beberapa contoh *stop-word* (dalam Bahasa Indonesia) diantaranya : yang, dari, di, hingga, ke, dan dengan. Konversi *term* ke bentuk dasar (*stemming*) juga merupakan tindakan yang dapat dilakukan dalam tahap ini. *Stemming* merupakan proses penghilangan *prefix* dan *suffix* dari kata untuk mendapatkan kata dasar (Grossmann, D & Frieder, O., 2004). Efektifitas algoritma *stemming* dapat diukur berdasarkan beberapa *parameter*, seperti proses, keakuratan, dan kesalahan.

3.2.3. Pattern Discovery

Pattern discovery merupakan tahap yang paling penting dalam proses *text mining*. Pada penelitian ini, operasi yang dilakukan adalah pengukuran tingkat kemiripan teks (*similarity*). Teknik yang digunakan pada tahap ini adalah dengan melakukan pembobotan (*weighting*)

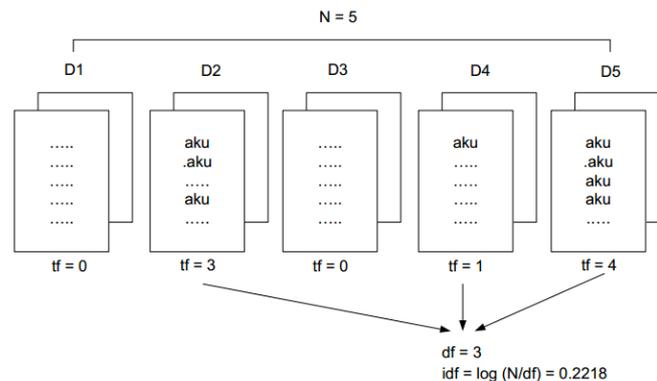
terhadap *term* dari hasil tahap *text transformation*. Setiap *term* diberikan bobot sesuai dengan skema pembobotan yang dipilih, baik itu pembobotan lokal, global, maupun kombinasi keduanya. Banyak aplikasi menerapkan pembobotan kombinasi berupa perkalian bobot lokal *term frequency* dan *global inverse document frequency*, ditulis $tf \times idf$. Dalam penelitian ini, akan menggunakan $tf \times idf$. Dalam kemiripan antar dokumen didefinisikan berdasarkan representasi *bag-of-words* dan dikonversi ke suatu model ruang vektor (*vector space model*). Setelah merepresentasi term ke dalam bentuk vektor, akan dilakukan proses pencarian kemiripan dengan menghitung nilai *cosinus* antar vektor yang dalam penelitian ini menggunakan *cosine similarity*.

3.3. Term Frequency (tf) – Inverse Document Frequency (idf)

Term Frequency (tf) merupakan frekuensi kemunculan suatu kata (*term*) dalam dokumen. Oleh sebab itu, *tf* memiliki nilai yang bervariasi dari satu dokumen ke dokumen lain tergantung dari tingkat kepentingan sebuah *term* dalam sebuah dokumen. Semakin sering suatu *term* muncul dalam suatu dokumen, *term* tersebut akan memiliki nilai *tf* yang lebih besar daripada *term-term* lain yang jarang muncul.

Penggunaan faktor *tf* belum mencukupi dalam menentukan pembobotan. Untuk itu, diperlukan faktor *Inverse Document Frequency* (*idf*) yang merupakan sebuah statistik “global” yang mengkarakteristikan sebuah *term* dalam keseluruhan koleksi dokumen. *idf* merupakan sebuah perhitungan dari bagaimana *term* didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Semakin sedikit dokumen yang mengandung *term* yang dimaksud, maka nilai *idf* semakin *besar*. Jika setiap dokumen dalam koleksi mengandung *term* yang bersangkutan, maka nilai *idf* dari term tersebut adalah nol (0). Hal ini menunjukkan bahwa setiap *term* yang muncul pada dokumen dalam koleksi tidak berguna untuk membedakan dokumen berdasarkan topik tertentu.

Ilustrasi algoritma *tf-idf* (Harlian, M., 2006) ditunjukkan pada gambar berikut :



Gambar 1. Ilustrasi algoritma *tf-idf*

Keterangan :

D1 - D5 = dokumen

tf = banyaknya term yang dicari pada setiap dokumen

N = total dokumen

Df = banyaknya dokumen yang mengandung term yang dicari

Dalam penelitian ini, algoritma pembobotan *Term Frequency* (*tf*) – *Inverse Document Frequency* (*idf*) ditetapkan pada tahap *similarity document*. Nilai *tf-idf* diperoleh dengan menggunakan persamaan (Lee, D.L., Chuang, H., Seamons, K., 1997).

$$W_{i,j} = tf_{ij} \times idf_j = tf_{i,j} \times \log\left(\frac{N}{df_j}\right) \dots \dots \dots (1)$$

Dimana :

$W_{i,j}$ = bobot term ke-j terhadap dokumen ke-i

tf_{ij} = jumlah kemunculan term j ke dalam dokumen i

N = jumlah dokumen secara keseluruhan

df_j = jumlah dokumen yang mengandung term j

Berdasarkan persamaan (1), berapapun besarnya nilai tf_{ij} apabila $N = df_j$ maka akan didapat hasil nol (0) untuk perhitungan idf . Untuk itu, dapat ditambahkan nilai 1 pada sisi idf , sehingga perhitungan bobotnya menjadi :

$$w_{ij} = tf_{i,j} \times \log\left(\frac{N}{df_j}\right) + 1 \dots\dots\dots(2)$$

Perhitungan bobot dari *term* tertentu dalam sebuah dokumen dengan menggunakan $tf \times idf$ menunjukkan bahwa deskripsi terbaik dari dokumen adalah *term* yang banyak muncul dalam dokumen tersebut dan sangat sedikit muncul pada dokumen lain (Lee, D.L., Chuang, H., Seamons, K., 1997).

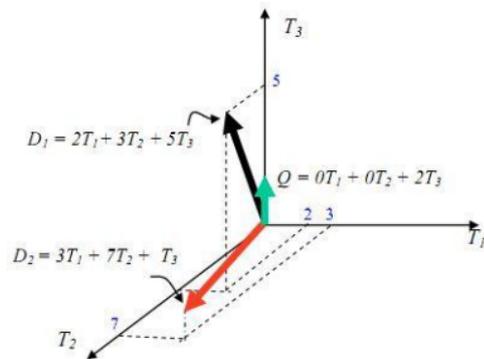
3.4. Vector Space Model

Model Ruang Vektor (*Vector Space Model*) merupakan suatu metode yang cukup banyak digunakan dalam sistem *Information Retrieval*. Dengan menggunakan Model Ruang Vektor, dokumen-dokumen yang ada akan direpresentasikan ke dalam n dimensi vektor.

Banyaknya dimensi dari ruang vektor akan ditentukan oleh jumlah kata signifikan yang terdapat dalam dokumen. Misalkan, terdapat sejumlah n kata yang berbeda sebagai kamus kata (*vocabulary*) atau indeks kata (*term index*). Kata-kata ini akan membentuk ruang vector yang memiliki dimensi sebesar n . setiap kata i dalam dokumen atau *query* diberikan bobot sebesar w_i . Baik dokumen maupun *query* direpresentasikan sebagai vektor berdimensi n . Contohnya, terdapat tiga buah kata, yaitu t_1, t_2, t_3 . Lalu ada dua buah dokumen D_1 dan D_2 serta sebuah *query* Q . masing-masing bernilai :

$$\begin{aligned} D_1 &= 2t_1 + 3t_2 + 5t_3 \\ D_2 &= 3t_1 + 7t_2 + 0t_3 \\ Q &= 0t_1 + 0t_2 + 2t_3 \end{aligned}$$

Ilustrasi penggambarannya kira-kira sebagai berikut :



Gambar 2. Ilustrasi vector space model

Koleksi dari n buah dokumen dapat direpresentasikan dalam Model Ruang Vektor dengan sebuah matriks *term-document* (Mandala, Rila, 2006). Berikut penggambaran matriksnya :

Document space						
	↓	t_1	t_2	t_3	$t..$	t_n
					←	Term vector space
D_1		w_{11}	w_{12}	w_{13}	$..$	w_{1n}
D_2		w_{21}	w_{22}	w_{23}	$..$	w_{2n}
$D..$		$..$	$..$	$..$	$..$	$..$
D_n		w_{m1}	w_{m2}	w_{m3}	$..$	w_{mn}

Gambar 3. Ilustrasi matriks vector space model

3.5. Matching Documents

Matching Documents adalah tahap dimana akan dilakukan perhitungan tingkat kemiripan (*similarity*) antara dokumen acuan (kunci jawaban) dengan kandidat dokumen (jawaban siswa). Perhitungan *matching documents* ini menggunakan metode *Cosine Similarity*. Rumus dari *Cosine Similarity* itu sendiri adalah sebagai berikut :

$$Sim(D, D_i) = \cos\theta = \frac{D \cdot D_i}{|D||D_i|} = \frac{\sum w_{q,j}w_{i,j}}{\sqrt{\sum w_{q,j}^2} \sqrt{\sum w_{i,j}^2}}$$

Dimana :

D = Dokumen acuan

D_i = Dokumen ke-i

$W_{q,j}$ = Bobot term j terhadap dokumen acuan

$W_{i,j}$ = Bobot term j terhadap dokumen ke-i

Similarity atau $Sim(D, D_i) = 1$ jika $D = D_i$, dan $Sim(D, D_i) = 0$ ketika D dan D_i tidak memiliki kesamaan sama sekali (benar-benar berbeda).

3.6. Metode Nazief-Andriani

Algoritma *Stemming* untuk bahasa yang satu berbeda dengan algoritma *Stemming* untuk bahasa yang lainnya. Sebagai contoh, bahasa Inggris memiliki morfologi yang berbeda dengan Bahasa Indonesia, sehingga algoritma *Stemming* untuk kedua bahasa tersebut juga berbeda. Proses *Stemming* pada teks berbahasa Indonesia lebih rumit dan lebih kompleks, karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan kata dasar dari sebuah kata.

Pada umumnya, kata dasar pada Bahasa Indonesia terdiri dari kombinasi :

Prefiks 1 + Prefiks 2 + Kata Dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

Algoritma Nazief-Andriani yang dibuat oleh Bobby Nazief dan Mirna Andriani ini memiliki tahap-tahap sebagai berikut :

1. Pertama, cari kata yang akan di-*stem* dalam Kamus Besar Bahasa Indonesia. Jika ditemukan maka asumsikan kata sebagai *root word*. Algoritma berhenti.
2. Inflection Suffixes (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika ada yang berupa partikel (“-lah”, “-kah”, “-tah”, atau “-pun”) maka langkah ini diulang lagi untuk menghapus Possessive Pronouns (“-ku”, “-mu”, atau “-nya”) jika ada.
3. Hapus Derivation Suffixes (“-i”, “-an”, atau “-kan”). Jika kata ditemukan di kamus, algoritma berhenti. Jika tidak, maka akan dilanjutkan ke langkah 3a :
 - a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k” maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus, maka algoritma berhenti. Jika masih tidak ditemukan, maka lakukan langkah 3b.
 - b. Akhiran yang dihapus (“-i”, “-an”, atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hilangkan derivation prefixes (DP) {“di-”, “ke-”, “se-”, “me-”, “be-”, “pe-”, “te-”} dengan iterasi maksimum tiga kali.
 - a. Langkah 4 berhenti jika :
 - Terjadi kombinasi awalan dan akhiran yang terlarang seperti pada Tabel 1.
 - Awalan yang dideteksi saat ini sama dengan awal yang dihilangkan sebelumnya.
 - Tiga awalan telah dihilangkan.

Tabel 1. Kombinasi Awalan-Akhiran yang tidak diizinkan

Awalan	Akhiran yang tidak diizinkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an

b. Identifikasi tipe awalan dan hilangkan. Awalan ada tipe :

- Standar : “di-”, “ke-”, “se-” yang dapat langsung dihilangkan dari kata.
- Kompleks : “me-”, “be-”, “pe-”, “te-” adalah tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Oleh karena itu, gunakan aturan pada Tabel 2 untuk mendapatkan pemenggalan yang tepat.

Tabel 2. Pemenggalan Awalan Stemmer Nazief dan Adriani

Aturan	Format Kata	Pemenggalan
1.	berV...	ber-V... be-rV...
2.	berCAP...	berCAP... dimana C!=’r’ & P!=’er’
3.	berCAerV...	ber-CAerV... dimana C!=’r’
4.	belajar	bel-ajar
5.	beC ₁ er C ₂ ...	be-C ₁ er C ₂ ... dimana C!=’r’ ’1’}
6.	terV...	ter-V... te-rV..
7.	terCerV...	ter-CerV... dimana C!=’r’
8.	terCP...	ter-CP... dimana C!=’r’
9.	teC ₁ er C ₂ ...	te-C ₁ er C ₂ ... dimana C!=’r’
10.	me{ r w y}V...	me-{ 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 C!=’r’
20.	pe{w y}V...	pe-{ w y}V...
21.	perV...	perV... pe-rV...
22.	perCAP...	per-CAP... dimana C!=’r’ dan P!=’er’
23.	perCAerV...	per-CAerV... dimana C!=’r’
24.	pem{b f V}...	pem-{ b f V}...
25.	pem{rV V}...	pe-m{rV V}... pe-p{rV V}...
26.	pen{c d j z}...	pen-{ c d j z}...
27.	penV...	pe-nV... pe-tV...
28.	peng{g h q}...	peng-{ g h q}...
29.	pengV...	peng-V... peng-kV...
30.	penyV...	peny-sV...
31.	peIV...	pe-IV... kecuali “pelajar” yang menghasilkan kata “ajar”
32.	peCerV..	per-erV.. dimana C!=’r w y l m n’
33.	peCP..	pe-CP.. dimana C!=’r w y l m n’

c. Cari kata yang telah dihilangkan awalnya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 4 diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan.

5. Apabila setelah langkah ke-4 kata dasar masih belum ditemukan, maka proses recording dilakukan dengan mengacu pada Tabel x. Recoding dilakukan dengan menambahkan karakter recoding awal kata yang dipenggal. Pada Tabel x, karakter recoding adalah huruf kecil setelah tanda hubung (-) dan terkadang berada di dalam tanda kurung ({}).
6. Jika semua langkah telah selesai tetapi tidak juga berhasil, maka kata awal diasumsikan sebagai root word, dan proses selesai.

Tipe awalan ditentukan melalui langkah-langkah sebagai berikut:

1. Jika awalnya adalah : “di-”, “ke-”, atau “se-”, maka tipe awalnya secara berturut-turut adalah : “di-”, “ke-”, atau “se-”.

2. Jika awalnya adalah : “te-”, “me-”, “be-”, atau “pe-” maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalnya.
3. Jika dua karakter pertama bukan “di-”, “ke-”, “se-”, “te-”, “me-”, “be-”, atau “pe-” maka berhenti.
4. Jika tipe awalan adalah “none” maka berhenti. Jika tipe awalan adalah bukan “none” maka awalan dapat dilihat pada Tabel 3. Hapus awalan jika ditemukan.

Tabel 3. Cara Menentukan Tipe Awalan Untuk awalan “te”

Following Characters				Tipe Awalan
Set 1	Set 2	Set 3	Set 4	
“-r-”	“-r-”	-	-	None
“-r-”	-	-	-	Terluluh
“-r-”	not (vowel or “-r-”)	“-er-”	Vowel	ter
“-r-”	not (vowel or “-r-”)	“-er-”	not vowel	ter-
“-r-”	not (vowel or “-r-”)	not “-er-”	-	ter
not (vowel or “-r-”)	“-er-”	vowel	-	None
not (vowel or “-r-”)	“-er-”	not vowel	-	None

Tabel 4. Jenis Awalan Berdasarkan Tipe Awalannya

Awalan	Akhiran yang harus dihapus
di-	di-
ke-	ke-
se-	se-
te-	te-
ter-	ter-
ter-luluh	Ter

Untuk mengatasi keterbatasan pada algoritma di atas, maka ditambahkan aturan-aturan di bawah ini:

1. Aturan untuk duplikasi.
 - a) Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka *root word* adalah bentuk tunggalnya, contoh : “buku-buku” *root word*-nya adalah “buku”.
 - b) Kata lain, misalnya “bolak-balik”, “berbalas-balasan”, dan “seolah-olah”. Untuk mendapatkan *root word*-nya, kedua kata diartikan secara terpisah. Jika keduanya memiliki *root word* yang sama, maka diubah menjadi bentuk tunggal, contoh : kata “berbalas-balasan”, “berbalas” dan “balasan” memiliki *root word* yang sama, yaitu “balas”, maka *root word* “berbalas-balasan” adalah “balas”. Sebaliknya, pada kata “bolak-balik”, “bolak” dan “balik” memiliki *root word* yang berbeda, maka *root word*-nya adalah “bolak-balik”.

2. Tambahan bentuk awalan dan akhiran serta aturannya.
 - a) Untuk tipe awalan “mem-”, kaya yang diawali dengan awalan “memp-” memiliki tipe awalan “mem-”.
 - b) Tipe awalan “meng-”, kata yang diawali dengan awalan “mengk-” memiliki tipe awalan “meng-”.

4. METODOLOGI

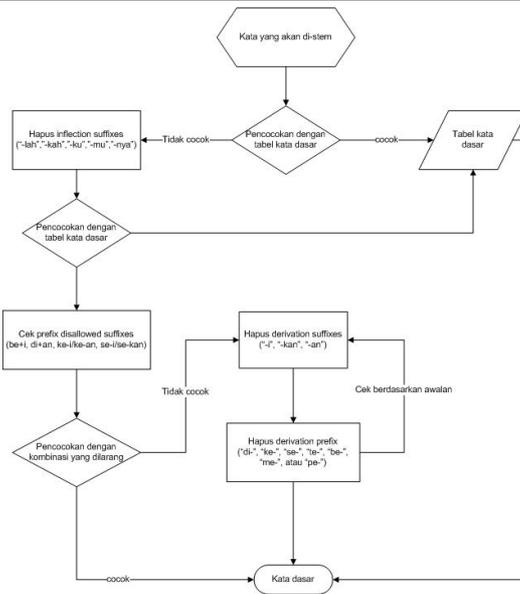
4.1 Bentuk Metodologi Penelitian

Metodologi penelitian yang digunakan untuk menjawab penelitian ini adalah metode penelitian kualitatif, dimana metodologi kualitatif merupakan penelitian yang bermaksud untuk memahami fenomena tentang apa yang dialami oleh subjek penelitian secara holistik (utuh) dengan cara deskripsi dalam bentuk kata-kata dan bahasa pada suatu konteks khusus yang alamiah (Moeloeng, Lexy J, 2004). Metode yang digunakan meliputi :

1. Pengumpulan Data
Mengumpulkan data soal mata pelajaran Geografi tingkat SMA yang akan diujikan dari berbagai sumber, serta mengumpulkan konsep-konsep atau teori-teori yang menunjang dalam penelitian ini.
2. Pemodelan Ujian Essay
Menentukan kata kunci yang tepat dari kunci jawaban sebagai acuan dalam penilaian ujian, memeriksa jawaban dari siswa dengan menjadikan kata kunci sebagai rujukan jawaban yang benar, menghitung semua nilai hasil dari perhitungan bobot masing-masing soal dan dijumlahkan dengan hasil kali nilai maksimal setiap pada setiap soal yang nantinya akan menjadi nilai akhir siswa.
3. Konstruksi Ujian Essay
Menentukan metode yang digunakan untuk mencocokkan kunci jawaban dengan jawaban dari siswa ke dalam sistem serta melakukan pembenahan fungsi ketika terjadi *error*.
4. Modifikasi LMS dengan fitur ujian essay
Memodifikasi LMS Moodle dengan menambahkan fungsi *text similarity* dan penilaian berdasarkan *text similarity* dalam ekstensi ujian essay serta membuat soal ujian yang akan diujikan ke dalam sistem.
5. Pengujian
 - a. Pengujian Struktur
Tahap pencocokan jawaban antara jawaban siswa dengan kunci jawaban menggunakan metode *text similarity* yang disertai hasil dari proses pengolahan.
 - b. Pengujian Pakar
Pencocokan jawaban antara jawaban siswa dengan kunci jawaban berdasarkan relevansi jawaban menurut pakar, yang dalam penelitian ini adalah guru Geografi SMA.

4.2 Pembuatan *stemming* menggunakan algoritma Nazief-Adriani

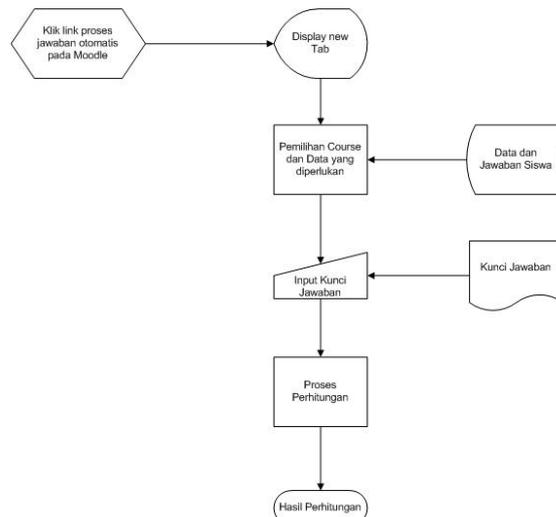
Pembuatan sistem penilaian otomatis menggunakan *text similarity* mempunyai beberapa tahap, yaitu *tokenizing*, *filtering*, *stemming*, pembobotan kata dalam dokumen menggunakan metode *tf-idf*, serta penilaian menggunakan metode *cosine similarity*. Diantara proses-proses tersebut, yang membedakan dengan *text similarity* yang lain adalah pada proses *stemming*. Proses ini merupakan proses mengubah kata menjadi kata dasar. Proses *stemming* yang digunakan pada sistem ini menggunakan algoritma Nazief-Adriani dikarenakan bahasa yang digunakan pada sampel ujian adalah bahasa Indonesia. Berikut ini merupakan alur dari algoritma *stemming* Nazief-Adriani yang dibuat.



Gambar 4. Flowchart sistem stemming berdasarkan algoritma Nazief-Adriani

4.3 Pembuatan Sistem

Setelah membuat proses stemming, lalu menggabungkan semua proses (*tokenizing, filtering, stemming, tf-idf, cosine similarity*), lalu dibuatlah perancangan sistem penilaian seperti di bawah ini:



Gambar 5. Flowchart proses sistem penilaian otomatis

5. HASIL DAN PEMBAHASAN

Proses ujicoba menggunakan tiga puluh sampel dengan sepuluh soal yang berbeda, lalu hasil penilaian dari pakar dibandingkan dengan penilaian dari sistem menggunakan uji T berpasangan (*T-test paired*) untuk mengetahui perbedaan nilai signifikan diantara keduanya dengan derajat bebas (α) sebesar 5% serta menggunakan hipotesis awal (H_0) adalah kedua nilai tidak berbeda signifikan sedangkan hipotesis alternatif (H_1) berarti kedua nilai berbeda signifikan. Berikut hasil perbandingan hasil akhir tersebut :

Tabel 5. Hasil uji-T nilai akhir penilaian pakar dengan penilaian sistem

	<i>Nilai Pakar</i>	<i>Nilai Sistem</i>
Mean	82.11667	81.88633
Variance	441.0118	399.4997
Observations	30	30
Pearson Correlation	0.987776	
Hypothesized Mean Difference	0	
Df	29	
t Stat	0.375501	
P(T<=t) two-tail	0.71002	
t Critical two-tail	2.04523	

Pada tabel 5, diketahui bahwa nilai t hitung (*t Stat*) adalah 0.3755, nilai signifikansi uji dua pihak (*P(T<=t) two-tail*) adalah 0.71, dan nilai t tabel uji dua pihak (*t Critical two-tail*) adalah 2.045. Berdasarkan data tersebut maka :

$t_{hitung} = 0.3755$ dan $t_{tabel} = 2.045$ sehingga $t_{tabel} > t_{hitung} \rightarrow H_0$ diterima

Nilai signifikansi = $0.71 > \alpha = 0.05 \rightarrow H_0$ diterima

Dengan diterimanya H_0 , ini berarti hasil penilaian sistem tidak berbeda signifikan dengan hasil penilaian pakar, sehingga sistem penilaian otomatis dapat diaplikasikan untuk ujian bertipe uraian (*essay*).

Namun, terdapat permasalahan ketika nilai dari masing-masing soal diuji dengan cara yang sama. Berikut tabel hasil uji T yang dilakukan terhadap penilaian dari masing-masing soal :

Tabel 6. Hasil uji-T penilaian pakar dengan penilaian sistem untuk masing-masing soal

Kode Soal	T Stat	P(T<=t) two-tail	t Critical two-tail	uji t	uji P
Q1	0.7108	0.4828	2.045	H0 diterima	H0 diterima
Q2	7.4638	3.1679E-08	2.045	H1 diterima	H1 diterima
Q3	0.7791	0.4422	2.045	H0 diterima	H0 diterima
Q4	7.9385	9.3644E-09	2.045	H1 diterima	H1 diterima
Q5	9.0712	5.7419E-10	2.045	H1 diterima	H1 diterima
Q6	-10.0259	6.2239E-11	2.045	H1 diterima	H1 diterima
Q7	-0.8931	0.3791	2.045	H0 diterima	H0 diterima
Q8	-8.7384	1.2814E-09	2.045	H1 diterima	H1 diterima
Q9	-1.1148	0.2741	2.045	H0 diterima	H0 diterima
Q10	7.0973	8.273E-08	2.045	H1 diterima	H1 diterima

Dari tabel 6, diketahui bahwa terdapat beberapa soal yang memiliki perbedaan penilaian yang signifikan, yaitu pada soal Q2, Q4, Q5, Q6, Q8, dan Q10.

Pada Q2 memiliki hasil penilaian yang berbeda signifikan dikarenakan jawaban yang terrekam di dalam *database* memiliki beberapa karakter yang tidak lazim. Karakter ini terrekam dikarenakan banyaknya siswa yang menggunakan *tinymce* untuk menguraikan jawaban mereka, sehingga jawaban mereka terrekam bersamaan dengan karakter unik *tinymce* ke dalam *database*.

Sedangkan pada Q4, Q5, Q6, Q8, dan Q10, memiliki hasil yang berbeda signifikan dikarenakan jawaban pada kunci tidak memiliki *keyword* yang dijadikan patokan sebagai jawaban yang benar, sehingga kata-kata yang sama antara kunci jawaban dan jawaban siswa, namun tidak memiliki makna akan dianggap sebagai jawaban yang benar berdasarkan penilaian dari sistem.

6. PENUTUP

Kesimpulan yang dapat diambil dari hasil pengujian adalah sebagai berikut :

1. Penilaian menggunakan *text similarity* dengan metode *tf-idf* menghasilkan output yang sesuai dengan spesifikasi dari user, namun membutuhkan waktu komputasi yang cukup lama ketika data yang diproses (teks dokumen) jumlahnya banyak.
2. Kata-kata di dalam kunci jawaban sangat berpengaruh terhadap pemilihan kata yang dianggap sebagai *keyword*, sehingga mempengaruhi penilaian dari sistem.
3. Hasil akhir penilaian menggunakan sistem *text similarity* tidak terlalu berbeda signifikan dibandingkan dengan penilaian berdasarkan pakar, namun harus diperhatikan dalam pemilihan jawaban yang akan dijadikan kunci agar tidak menjadi kata-kata yang tidak mengandung arti (bias).

Saran yang dapat diberikan dari hasil pengujian diantaranya :

1. Untuk kunci jawaban, diusahakan memberikan jawaban yang sangat berkaitan dengan kata kunci yang dimaksud dan tidak terlalu banyak kata-kata yang tidak mengandung arti (bias).
2. Sistem ini dapat dikembangkan, disempurnakan, dan dibandingkan menggunakan metode yang lain.

DAFTAR PUSTAKA

- Agusta, Ledi (2009). *Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia*. Konverensi Nasional Sistem dan Informatika 2009.
- Alotaiby, F.T.; & Chen, J. X. (2005). "Generic Summative Assessment Functional Model". Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05). IEEE. 1-6.
- Dorre, J, Gerstl, P & Seiffert, R. (1999). *Text Mining : Finding Nuggets in Mountains of Textual Data*. SQSD IDM, Germany.
- Grossmann, D & Frieder, O. (2004). *Information Retrieval : Algorithm And Heuristic, Second Edition*. Springer, Dordrecht, The Netherlands.
- Harlian, M. (2006). *Machine Learning Text Categorization*. University of Texas, Austin.
- Hidayat, Arief.; & Sugiharto, Aris (2012). *Implementasi Arsitektur Sistem Adaptif Dalam Rancang Bangun Sistem Kuis Online*. Seminar Nasional Aplikasi Teknologi Informasi (SNATI) 2012, Yogyakarta, 15-16 Juni 2012.
- Kurniawan, Bambang., Effendi, Syahril., Sitompul, Opim Salim. (2012). *Klasifikasi Konten Berita dengan Metode Text Mining*. Jurnal Dunia Teknologi Informasi Vol. 1, No. 1 (2012) hal. 14-19.
- Lalos, P., Retalis, S. & Psaroniligkos, Y. (2005). *Creating Personalised Quizzes Both to the Learner and to the Access Device Characteristics: the Case of CosyQTI*. A3EH: Third International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia July 19, 2005 at AIED'05.
- Lee, D.L., Chuang, H., Seamons, K., (1997). *Document Ranking and the Vector Space Model*. IEEE Software, vol. 14, Issue 2, pp. 67 – 75.
- Mandala, Rila. (2006). *Evaluasi Kinerja Sistem Penyaringan Informasi Model Ruang Vektor*. Institut Teknologi Bandung.

- Moeloeng, Lexy J. "Metode Penelitian Kualitatif". Remaja Rosda Karya, Bandung. 2004.
- Muhajir, Riky Bagus. (2012). *Metode Similarity-Mashup untuk Framework Modul Relevant Content pada Content Management system (CMS)*. Jurusan Informatika – Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret, Surakarta.
- Negara, Sabda R.A. (2011), *Pembuatan Sistem Uji Kompetensi Pelatihan Online*. Politeknik Elektronika Negeri Surabaya, Institut Teknologi Sepuluh Nopember, Surabaya.
- QuestionMark; & League. (2004). *An Assessment Framework for the Community College*. Diakses pada 6 Maret 2013 dari <http://www.league.org/publication/whitepapers/files/0804.pdf>.
- Quinn, D.; & Reid, I. 2003. "Using Innovative Online Quizzes to Assist Learning". AusWeb03 : changing the way we work : proceedings of AusWeb03, the ninth Australian World Wide Web Conference, 2003 (ISBN: 1863844937).
-