

## **Aplikasi Penilaian Otomatis Jawaban Soal *Essay* dengan Menggunakan Metode *K-Nearest Neighbors***

### ***Application of Automatic Assessment of Essay Questions by Using the K-Nearest Neighbors Method***

**Ni Nyoman Emang Smrti<sup>1</sup>, Aggry Saputra<sup>2</sup>, Zulfachmi<sup>3</sup>, Ricak Agus Setiawan<sup>4</sup>  
I Putu Gd. Sukenada Andisana<sup>5</sup>, I Putu Yogi Pratama Putra<sup>6</sup>**

<sup>1,2,3,4,5,6</sup> STMIK Bandung Bali

Email: smrti@yacanet.com, aggrysaputra@gmail.com, fahmi.alfahim.93@gmail.com,  
ricak.agus@gmail.com, sukenada.andisana@gmail.com, youputra@gmail.com

#### **Abstrak**

Evaluasi proses belajar mengajar dilakukan dengan memberikan soal baik dengan bentuk *essay* maupun pilihan ganda. Proses koreksi untuk soal pilihan ganda relative lebih mudah, karena jawabannya hanya satu dan pasti. Proses koreksi pada Jawaban soal dalam bentuk *essay*, memerlukan waktu yang lebih lama dibandingkan dengan jawaban soal pilihan ganda, karena jawaban *essay* sangat bervariasi. Penilaian secara otomatis untuk jawaban *essay*, tidak dapat bergantung dari hasil pencocokan jawaban dengan kunci jawaban, karena memungkinkan jawabannya memiliki makna yang sama namun pemilihan katanya yang berbeda. Dengan asumsi bahwa ujian dilakukan dengan menggunakan komputer atau e-learning, maka koreksi jawaban soal *essay* akan dibuat otomatis. Penelitian ini disamping mencocokkan jawaban dengan kunci jawaban juga memanfaatkan penilaian terdahulu sebagai data latih yang akan digunakan sebagai dasar untuk penilaian saat ini. Metode yang digunakan untuk menilai jawaban *essay* adalah K-NN dengan menganggap K sama dengan 1 dibuatlah program aplikasi pencocokan hasil ujian. Penelitian ini mengambil data hasil ujian siswa SMP PGRI 3 Denpasar untuk mata pelajaran TIK. Dari hasil pengujian didapatkan bahwa tingkat akurasi yang tertinggi diperoleh 100% dan rata-rata akurasinya adalah 97,6%.

Kata Kunci: K-Nearest Neighbors, penilaian otomatis, TF-IDF

#### **Abstract**

*Evaluation of teaching and learning process is done by giving the problem either with essay form or multiple choice. The correction process for multiple choice questions is relatively easy, because the answer is only one and for sure. The correction process on the Answer question in essay form takes longer time than the multiple choice answer, because the essay answers are very varied. Automatic assessment of the essay answers can not depend on the result of an answer match with the answer key, because it allows the answer to have the same meaning but different wording. Assuming that the test is done using a computer or e-learning, then the essay answer correction will be made automatically. This research in addition to matching answers with key answers also makes use of previous assessments as training data to be used as the basis for current assessments. The method used to evaluate the essay's answer is K-NN by assuming K equal to 1 is made the application program matching the exam results. This research takes data of student exam result of SMP PGRI 3 Denpasar for ICT subject. From the test results obtained that the highest level of accuracy is 100% and the average accuracy is 97.6%.*

*Keywords: K-Nearest Neighbors, Penilaian Otomatis, TF-IDF*

## 1. Pendahuluan

Ujian akhir semester merupakan salah satu alat evaluasi dalam proses belajar mengajar. Evaluasi proses pembelajaran dilakukan untuk menentukan kualitas pembelajaran secara keseluruhan.

Ujian akhir semester akan membuat kesibukan tambahan bagi guru terutama dalam koreksi. Proses koreksi soal pilihan ganda mungkin tidak terlalu susah, namun untuk soal *essay*, guru harus membutuhkan waktu yang agak lama, karena harus membaca satu persatu jawaban dari siswa tersebut. Penyelesaian koreksi hasil ujian tepat waktu sangatlah diharapkan oleh semua pihak, namun setiap guru juga memiliki kendala dalam proses koreksi, karena koreksi terkadang harus dibawa pulang, sedangkan di rumah para guru memiliki kewajiban yang tidak boleh ditinggalkan.

Penelitian terdahulu Sistem Koreksi Soal *Essay* Otomatis dengan Menggunakan Metode Rabin Karp proses koreksi jawaban *essay* biasanya hanya mengandalkan pencocokan jawaban dengan kunci jawaban (Hamza, Sarosa, & Santoso, 2013). Kekurangan dari sistem pencocokan ini adalah, apabila ada jawaban yang intinya sama namun pemilihan katanya tidak sama akan menghasilkan nilai yang tidak sesuai. Pada penelitian ini akan dibuatkan sistem yang memanfaatkan hasil koreksi guru terdahulu diinputkan menjadi data latih. Data latih inilah yang dijadikan dasar untuk mencocokkan jawaban siswa dan menghasilkan nilai yang lebih sesuai dibandingkan dengan bila jawaban siswa dicocokkan dengan kunci jawaban secara langsung.

*Text mining* memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antara dokumen (Kusrini & Luthfi, 2009). *Text mining* dapat didefinisikan secara luas sebagai suatu proses menggali informasi dimana seorang *user* berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen-komponen dalam data mining yang salah satunya adalah kategorisasi. *Text mining* bisa dianggap subjek riset yang tergolong baru. *Text mining* dapat memberikan solusi dari permasalahan seperti pemrosesan, pengorganisasian/ pengelompokkan dan menganalisa *unstructured text* dalam jumlah besar (Feldman & Sanger, 2009).

Setelah didapatkan kata-kata dari hasil jawaban masing-masing siswa, langkah selanjutnya adalah mencocokkannya dengan kunci jawaban dan data latih. Pencocokan jawaban siswa dengan kunci jawaban dan data latih akan digunakan metode K-NN.

## 2. *Text Mining* dan Atributnya

### A. *Preprocessing*

*Text mining* merupakan proses mengesktrak *patterns* dan *knowledge* yang bersifat menarik dan penting dari dokumen-dokumen teks. Pada intinya proses kerja *text mining* sama dengan proses kerja data mining pada umumnya hanya saja data yang di mining merupakan *text databases* (Kusrini & Luthfi, 2009). Sebelum proses pencocokan jawaban, maka kunci jawaban harus melalui proses *preprocessing* yaitu melalui tahapan *text mining*. Tahap-tahap *text mining* secara umum adalah:

1. Tahap *tokenizing* adalah tahap pemotongan *string* input berdasarkan tiap kata yang menyusunnya.
2. Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil token. Algoritma yang digunakan adalah algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting).
3. Tahap *stemming* adalah tahap mencari *root* kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Tahap ini kebanyakan dipakai untuk teks berbahasa Inggris dan lebih sulit diterapkan pada teks berbahasa Indonesia.
4. Tahap *tagging* adalah tahap mencari bentuk awal/*root* dari tiap kata hasil *stemming*.

## B. TF-IDF

Tahap *analyzing* merupakan tahap penentuan seberapa jauh keterhubungan antara kata-kata antar dokumen yang ada. Tahap ini menggunakan algoritma frekuensi *term* (TF), *inverse document frequency* (IDF) dan kombinasi perkalian antara keduanya (TFxIDF).

Rumus pembobotan dasar pencarian dasar yang dikenal dengan IDF, diusulkan oleh Sparck Jones pada heuristik pada tahun 1972, telah terbukti sangat kuat (Robertson, 2004).

Metode ini akan menghitung nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) pada setiap token (kata) di setiap dokumen dalam korpus (Maarif, 2015). TFIDF ini akan menghitung bobot setiap token  $t$  di dokumen  $d$  dengan rumus:

$$W_{dt} = tf_{dt} * IDF_t \quad (1)$$

Dimana:

$d$  : dokumen ke- $d$   $t$  : kata ke- $t$  dari kata kunci

$W$  : bobot dokumen ke- $d$  terhadap kata ke- $t$

$tf$  : banyaknya kata yang dicari pada sebuah dokumen IDF

*Inversed Document Frequency* Nilai IDF didapatkan dari

$$IDF = \log_2\left(\frac{D}{df}\right) \quad (2)$$

Dimana:

$D$  : total dokumen

$df$  : banyak dokumen yang mengandung kata yang dicari

## C. Algoritma Nazief dan Adriani

Algoritma ini merupakan algoritma *stemming* berbahasa Indonesia yang dibuat oleh Bobby Nazief dan Mirna Adriani.

Berdasarkan penelitian Manase Sahat H Simarankir Algoritma Nazier dan Adriani memiliki hasil evaluasi yang lebih tinggi. Hal ini dikarenakan pada Algoritma Nazief dan Adriani terdapat penambahan aturan-aturan seperti penambahan aturan untuk reduplikasi, penambahan aturan untuk awalan dan akhiran dalam meningkatkan presisi dari setiap kata yang di *stemming* (Sahat H. Simarankir, 2013).

Algoritma ini menggunakan beberapa aturan morfologi untuk menghilangkan *affiks* (awalan, imbuhan dan sisipan) dari sebuah kata dan kemudian mencocokkannya dalam *database* kata dasar. Algoritma ini memiliki tahapan:

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tersebut adalah *root word* Maka algoritma berhenti.
2. *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa *particles* (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus *Derivation Suffixes* (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka 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 tidak ditemukan maka lakukan langkah 3b.
  - b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus *Derivation Prefix*. Jika pada langkah 3 ada *sufiks* yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
  - a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
  - b. For  $i = 1$  to 3, tentukan tipe awalan kemudian hapus awalan. Jika *root word* belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.

5. Melakukan *Recoding*.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

**D. K-Nearest Neighborhood (K-NN)**

Algoritma *K-nearest neighborhood* (K-NN) merupakan algoritma *supervised learning* di mana hasil klasifikasi data baru berdasar kepada kategori mayoritas tetangga terdekat ke-K. Tujuan dari algoritma ini adalah mengklasifikasikan objek baru berdasarkan atribut dan data *training*. Algoritma K-NN menggunakan klasifikasi ketetangga sebagai prediksi terhadap data baru (Gorunescu, 2011).

Nilai k yang terbaik untuk algoritma ini tergantung pada data, pada umumnya nilai k yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*.

Langkah-langkah Algoritma K-nearest neighbors (K-NN):

1. Tentukan Parameter K = jumlah tetangga terdekat.
2. Hitung jarak antara data yang akan ditentukan klasifikasinya dengan semua sampel pelatihan.
3. Urutkan jarak dan tentukan tetangga terdekat berdasarkan jarak minimum K.
4. Kumpulkan kategori tetangga terdekat.

**3. Metodologi**

Bab ini membahas analisa, perancangan sistem, gambaran umum sistem dan implementasi.

**A. Analisa Sistem**

Hasil dari analisa ada tiga *user* yang dapat menggunakan sistem ini yakni *user administrator*, *user guru* dan *user siswa*. *User administrator* dapat memasukan data siswa, data guru, data kelas dan data mata pelajaran ke dalam sistem. Data yang diinputkan oleh administrator ini akan digunakan pada proses pengelolaan data mata pelajaran yang diujikan serta data siswa sebagai peserta ujian. *User guru* dapat memasukkan data soal dan data latihan. *User siswa* dapat mengakses data soal, kemudian menjawab dan mendapatkan hasil berupa nilai.

**B. Perancangan Sistem**

Sistem dirancang untuk dapat membantu guru dalam melakukan proses koreksi jawaban siswa khususnya pada soal esai. Sebelum melakukan proses koreksi sistem ini harus diisi dengan data latihan. Data latihan terdiri dari soal, beberapa variasi jawaban disertai dengan nilai yang diberikan oleh guru. Data latihan ini diperoleh dari pengalaman guru dalam memberikan nilai dari jawaban siswa yang pernah dilakukannya. Ada 3 tahapan yang harus dilalui sebelum data jawaban siswa mendapatkan nilai yaitu:

1. Tahap pertama adalah *text mining*.
2. Tahap kedua adalah *analizing* yaitu perhitungan TFIDF. TFIDF adalah suatu tahap data teks dikonversi menjadi bobot.
3. Tahap ketiga perhitungan jarak atau kedekatan antara jawaban data latihan dengan jawaban siswa, sehingga diperoleh nilai yang sesuai dengan jawaban siswa.

Tahapan Pertama yaitu *text mining* akan diilustrasikan menggunakan data latihan berikut pada Tabel 1.

Tabel 1. Contoh Data Latihan

<b>Soal: Apa yang dimaksud dengan internet?</b>		
<b>Kata Kunci: kumpulan, komputer, terhubung, jaringan, pertukaran, informasi</b>		
<b>Data Latihan</b>		
#	Jawaban	Nilai

1	kumpulan komputer yang terhubung satu dengan lainnya dan membentuk sebuah jaringan	9
2	komputer yang berhubungan dan lokasinya berjauhan	8
3	pencarian informasi di google	6
4	perangkat yang terhubung satu dengan lainnya yang dapat bertukar informasi	8

Tahap Kedua yaitu TFIDF. Tahapan perhitungannya adalah sebagai berikut:

1. Inisialisasi (*Term Frequency*) banyaknya kata yang muncul di masing-masing *term*,
2. Pembobotan *term frequency*,
3. Pembobotan *inverse document frequency*,
4. Pembobotan dokumen TFIDF hingga menghasilkan bobot dari dokumen tersebut.

Pada Tabel 2 kata kumpul hanya muncul pada dokumen 1. Kata komputer muncul pada dokumen 1 dan 2. Kata hubung muncul pada dokumen 1,2, 4 dan 5. Kata satu muncul pada dokumen 1 dan 4. Kata bentuk muncul pada dokumen 1 dan 5. Kata sebuah hanya muncul pada dokumen 1. Kata jaring muncul pada dokumen 1 dan 5. Kata lokasi hanya muncul di dokumen 2. Kata jauh hanya muncul pada dokumen 2. Kata cari hanya muncul pada dokumen. Kata informasi muncul pada dokumen 3 dan 4. Kata google hanya muncul pada dokumen 3. Kata perangkat hanya muncul pada dokumen 4 dan 5. Kata tukar hanya muncul pada dokumen 4.

Dari contoh data latih diatas maka hasil dari perhitungan TFID nya dapat dilihat pada Tabel 2.

Tabel 2. Hasil perhitungan TFIDF

Term	TF						IDF	W				
	D1	D2	D3	D4	D5	Df		D1	D2	D3	D4	D5
Kumpul	1	0	0	0	0	1	0.7	0.699	0	0	0	0
komputer	1	1	0	0	0	2	0.4	0.398	0.398	0	0	0
hubung	1	1	0	1	1	4	0.1	0.097	0.097	0	0.097	0.097
satu	1	0	0	1	0	2	0.4	0.398	0	0	0.398	0
bentuk	1	0	0	0	1	2	0.4	0.398	0	0	0	0.398
sebuah	1	0	0	0	0	1	0.7	0.699	0	0	0	0
jaring	1	0	0	0	1	2	0.4	0.398	0	0	0	0.398
lokasi	0	1	0	0	0	1	0.7	0	0.699	0	0	0
jauh	0	1	0	0	0	1	0.7	0	0.699	0	0	0
cari	0	0	1	0	0	1	0.7	0	0	0.699	0	0
informasi	0	0	1	1	0	2	0.4	0	0	0.398	0.398	0
google	0	0	1	0	0	1	0.7	0	0	0.699	0	0
perangkat	0	0	0	1	1	2	0.4	0	0	0	0.398	0.398
tukar	0	0	0	1	0	1	0.7	0	0	0	0.699	0
<b>Total Weight</b>								<b>1.592</b>	<b>0.495</b>	<b>0.398</b>	<b>1.194</b>	<b>0.495</b>

Berdasarkan kata kunci yang telah didefinisikan diatas, bobot untuk masing-masing dokumen adalah sebagai berikut:

bobot (W) untuk D1 =  $0.699 + 0.398 + 0.097 + 0.398 + 0 + 0 = 1.592$

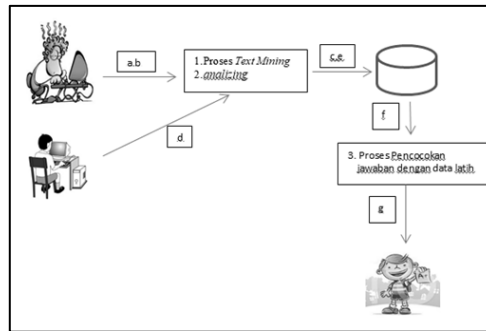
bobot (W) untuk D2 =  $0 + 0.398 + 0.097 + 0 + 0 + 0 = 0.495$

bobot (W) untuk D3 =  $0 + 0 + 0 + 0 + 0.398 + 0 = 0.398$

bobot (W) untuk D4 =  $0 + 0 + 0.097 + 0 + 0.398 + 0.699 = 1.194$

bobot (W) untuk D5 =  $0 + 0 + 0.097 + 0.398 + 0 + 0 = 0.495$

Perancangan sistem yang akan dibangun dapat dilihat pada Gambar 1.



Gambar 1. Gambaran Umum Sistem

Keterangan gambar:

1. Operator atau guru menginputkan data latih. Data latih adalah data soal, jawaban dengan berbagai variasi dan nilai disetiap jawaban yang berbeda.
2. Operator atau guru menginputkan soal beserta kunci jawaban.
3. Data Latih, Soal dan kunci jawaban yang diinputkan dimasukkan pada proses *text mining*. Setelah proses *text mining* dilanjutkan proses *analyzing* setelah selesai dilanjutkan dengan proses penyimpanan pada basis data. Soal sudah siap untuk disajikan.
4. Siswa menjawab soal. Jawaban siswa juga melalui proses *text mining* dan *stemming* kemudian disimpan dalam database.
5. Proses berikutnya adalah membandingkan jawaban siswa dengan data latih sehingga dapat diperoleh nilai.
6. Siswa dapat melihat nilai

#### 4. Pengujian

Pada penelitian ini pengujian yang dilakukan untuk mengetahui apakah metode klasifikasi K-NN ini dapat membantu memberikan nilai yang lebih mendekati dengan pemberian nilai yang diberikan oleh guru secara manual.

##### A. Uji Coba Metode K-NN

Uji coba ini dilakukan terhadap 5 jawaban soal dari 5 siswa. Hasil dari pengujian dapat dilihat pada Tabel 3 untuk  $k = 3$  dan Tabel 4 untuk  $k = 4$ .

Tabel 3. Hasil Pengujian Metode K-NN dengan  $K=3$ 

NIM	Jumlah Data Uji	Hasil	
		Benar	Salah
10207	5	3	2
10208	5	2	3
10209	5	5	0
10210	5	4	1
10211	5	3	2
Total		17	8

Berdasarkan Tabel 3 hasil pengujian metode K-NN dengan menggunakan  $k=3$  berjumlah 17 data yang benar dan 8 data yang salah.

Tabel 4. Hasil Pengujian Metode K-NN dengan  $K=4$ 

NIM	Jumlah Data Uji	Hasil	
		Benar	Salah
10207	5	3	2
10208	5	2	3
10209	5	5	0
10210	5	4	1

10211	5	4	1
Total		18	7

Berdasarkan Tabel 4 hasil pengujian metode K-NN dengan menggunakan k=4 berjumlah 18 data yang benar dan 7 data yang salah. Jika dibandingkan antara Tabel 3 dan Tabel 4, metode K-NN dengan menggunakan k=4 lebih tinggi kebenarannya daripada k=3. Akan tetapi untuk mendapatkan hasil perbandingan data, k=3 lebih efektif daripada k=4.

**B. Pengujian Terhadap banyaknya data latih dan penambahan kata kunci**

Pengujian dilakukan untuk memastikan implementasi berjalan sesuai dengan harapan. Sebelum dilakukan pengujian sistem, telah dipersiapkan data mata ujian, soal, kata kunci jawaban serta data latih dari sampel jawaban terdahulu. Soal yang digunakan dalam pengujian ini adalah soal TIK dengan jumlah soal 5 buah dimana masing-masing soal tersebut memiliki kata kunci jawaban serta data latih sebagai pembandingnya. Ada dua pengujian yang dilakukan dalam sistem ini.

1. Pengujian pertama adalah pengujian dengan menggunakan variasi jumlah data latih 10, 25 dan 50 data latih. Pengujian ini dilakukan untuk mengukur perbandingan tingkat akurasi dengan jumlah data latih yang digunakan. Sampel siswa yang digunakan berjumlah 5 orang. Dapat dilihat pada Tabel 5, 6 dan 7.
2. Pengujian kedua adalah pengujian untuk mengukur pengaruh dari penggunaan kata kunci jawaban terhadap akurasi penilaian. Pengujian ini menggunakan sampel 50 data latih dan 25 orang siswa. Dapat dilihat pada Tabel 8.

Berikut hasil pengujiannya:

**Tabel 5. Pengujian dengan 10 data latih**

NIM	Nilai Per Jawaban (Auto)					Total Nilai (Auto)	Nilai Per Jawaban (Manual)					Total Nilai (Manual)	Similaritas Nilai Jawaban (%)	Similaritas Total Nilai (%)	Total Similaritas (%)
	1	2	3	4	5		1	2	3	4	5				
10207	7	7	6	7	7	6.8	10	9	8	10	7	8.8	78.55555556	77.27272727	77.91414141
10208	7	7	6	8	7	7	10	10	7	9	8	8.8	80.42063492	79.54545455	79.98304473
10209	7	7	6	7	7	6.8	6	8	6	7	9	7.2	90.1984127	94.44444444	92.32142857
10210	7	7	6	7	7	6.8	7	7	9	8	6	7.4	87.97619048	91.89189189	89.93404118
10211	7	7	6	7	7	6.8	9	8	8	7	6	7.6	85.1984127	89.47368421	87.33604845
<b>Rata-Rata Similaritas</b>														85.49774087	

**Tabel 6. Pengujian dengan 25 data latih**

NIM	Nilai Per Jawaban (Auto)					Total Nilai (Auto)	Nilai Per Jawaban (Manual)					Total Nilai (Manual)	Similaritas Nilai Jawaban (%)	Similaritas Total Nilai (%)	Total Similaritas (%)
	1	2	3	4	5		1	2	3	4	5				
10207	8	8	6	8	7	7.4	10	9	8	10	7	8.8	84.77777778	84.09090909	84.43434343
10208	8	7	6	8	7	7.2	10	10	7	9	8	8.8	82.42063492	81.81818182	82.11940837
10209	7	7	6	7	8	7	6	8	6	7	9	7.2	92.42063492	97.22222222	94.82142857
10210	7	7	7	8	7	7.2	7	7	9	8	6	7.4	92.6984127	97.2972973	94.997855
10211	8	7	7	7	7	7.2	9	8	8	7	6	7.6	89.92063492	94.73684211	92.32873851
<b>Rata-Rata Similaritas</b>														89.74035478	

**Tabel 7. Pengujian dengan 50 data latih**

NIM	Nilai Per Jawaban (Auto)					Total Nilai (Auto)	Nilai Per Jawaban (Manual)					Total Nilai (Manual)	Similaritas Nilai Jawaban (%)	Similaritas Total Nilai (%)	Total Similaritas (%)
	1	2	3	4	5		1	2	3	4	5				
10207	10	9	7	10	7	8.6	10	9	8	10	7	8.8	97.5	97.72727273	97.61363636
10208	10	9	7	8	7	8.2	10	10	7	9	8	8.8	93.27777778	93.18181818	93.22979798
10209	6	7	6	7	8	6.8	6	8	6	7	9	7.2	95.27777778	94.44444444	94.86111111
10210	7	7	8	8	6	7.2	7	7	9	8	6	7.4	97.77777778	97.2972973	97.53753754
10211	8	7	8	7	6	7.2	9	8	8	7	6	7.6	95.27777778	94.73684211	95.00730994

Rata-Rata <i>Similaritas</i>	95.64987859
------------------------------	-------------

Dari hasil pengujian pertama yaitu pengujian untuk mengetahui pengaruh jumlah data latih terhadap hasil penilaian yang dapat dilihat pada Tabel 5, 6 dan 7 menunjukkan bahwa rata-rata *similaritas*/kemiripan dari nilai ujian yang dihasilkan oleh sistem dengan nilai ujian yang ditentukan secara manual. Pengujian penilaian dengan 10 data latih memiliki rata-rata *similaritas* sebesar 87.35%. Pengujian penilaian dengan 25 data latih memiliki rata-rata *similaritas* tingkat kemiripan sebesar 89.74% dan terakhir, pengujian penilaian dengan 50 data latih memiliki rata-rata *similaritas* tingkat kemiripan sebesar 95.65%.

Jadi dapat disimpulkan tingkat kemiripan antara nilai ujian yang dihasilkan oleh sistem dengan nilai ujian yang diperoleh secara manual berbanding lurus dengan jumlah data latih yang digunakan. Semakin banyak data latih yang digunakan dalam sistem ini maka semakin tinggi pula tingkat *similaritas* dari nilai ujian yang dihasilkan oleh sistem.

Tabel 8. Pengujian dengan 50 data latih

NIS Siswa	Nilai Ujian (Auto)	Nilai Ujian (Auto Tanpa Kata Kunci)	Nilai Ujian (Manual)	<i>Similaritas</i> Nilai Auto (%)	<i>Similaritas</i> Nilai Auto Tanpa Kata Kunci (%)	Perbedaan Persentase (%)
10207	8.6	6.8	9	95.6	75.6	20
10208	8.8	8	8.8	100	90.9	9.1
10209	7.6	6.2	7.6	100	81.6	18.4
10210	7.6	7.4	8	95	92.5	2.5
10211	8.6	7.6	9	95.6	84.5	11.1
10212	8.8	7.2	9	97.8	80	17.8
10213	9.2	5.6	8.8	95.6	63.6	32
10214	9	6.6	8.6	95.6	76.8	18.8
10215	8.2	6.2	8	97.6	77.5	20.1
10216	8.6	6.2	8	93	77.5	15.5
10217	8.8	8.2	9	97.8	91.2	6.6
10218	8.8	6.6	8.8	100	75	25
10219	9	6.8	8.8	97.8	77.2	20.6
10220	8	7.6	8	100	95	5
10221	8.6	8.2	8.6	100	95.4	4.6
10222	8.4	7.2	8.6	97.6	83.8	13.8
10223	9.2	7	9.2	100	76	24
10224	7.4	6	7.6	97.4	78.9	18.5
10225	7.8	6.8	7.8	100	87.2	12.8
10226	8.8	7.4	9	97.8	82.2	15.6
10227	8.4	6.2	8.6	97.6	72.1	25.5
10228	8.4	6.4	8.2	97.6	78.1	19.5
10229	8.8	6.8	9	97.8	75.6	22.2
10230	9	7.6	9.4	95.8	80.8	15
10231	8.2	6.8	7.8	95.2	87.2	8

Pengujian yang kedua yang terlihat pada Tabel 8 di atas adalah pengukuran tingkat akurasi penilaian dengan menggunakan bantuan kata kunci jawaban. Setiap soal yang ada akan disediakan kata kunci yang terkait dari masing-masing soal tersebut dan dalam pengujian ini akan dilakukan perhitungan dengan menggunakan bantuan dari kata kunci tersebut dan perhitungan tanpa menggunakan kata kunci sehingga kita akan mengetahui seberapa penting pengaruh kata kunci jawaban terhadap akurasi nilai jawaban.

Tingkat akurasi penilaian menggunakan metode K-NN dengan sampel 25 orang siswa dapat mencapai 93% yang terendah dan 100% yang tertinggi, jika perhitungan menggunakan kata kunci jawaban. Jika perhitungan nilai tidak menggunakan kata kunci jawaban, persentase akurasi penilaian dapat turun hingga 25,5%. Maka dari itu kata kunci jawaban pada suatu soal dianggap penting dalam sistem penilaian jawaban *essay* ini.



## 5. Kesimpulan dan Saran

Dari hasil pengujian maka dapat ditarik kesimpulan sebagai berikut:

1. Hasil dari pembobotan dokumen TFIDF tergantung dari hasil proses text mining dan kata kunci jawaban soal yang digunakan dalam dokumen tersebut. Bilamana sistem tidak menggunakan kata kunci jawaban maka akurasi penilaian turun hingga 25,5%.
2. Efektifitas dari penerapan metode K-Nearest Neighbors dalam penilaian jawaban esai otomatis berbanding lurus dengan jumlah data latih yang dipergunakan. KNN efektif digunakan untuk soal ujian yang telah memiliki data jawaban sebelumnya dengan variasi nilai yang beragam. Akan tetapi KNN tidak didesain untuk soal ujian baru yang belum sama sekali memiliki data jawaban terdahulunya.
3. Pengembangan selanjutnya yang harus dilakukan adalah apabila data latihnya semakin banyak akan mengakibatkan waktu yang diperlukan untuk melakukan proses klasifikasi akan bertambah lama. Saran untuk meminimumkan waktu proses klasifikasi yaitu dengan menggabungkan dengan metode *clustering*.

## Ucapan Terimakasih

Penulis mengucapkan rasa terimakasih yang sebesar-besarnya kepada pihak SMP PGRI 3 Denpasar yang telah memberikan kesempatan kepada penulis untuk melakukan penelitian dan juga kepada pihak STMIK Bandung Bali yang memberikan motivasi serta dukungan sehingga penelitian ini berjalan dengan lancar sebagaimana mestinya.

## Daftar Pustaka

- Hamza, S., Sarosa, M., & Santoso, P. B. (2013). Sistem Koreksi Soal *Essay* Otomatis Dengan Menggunakan Metode Rabin Karp. *Jurnal EECCIS*, Vol. 7 No. 2 pp. 153–158.
- Kusrini & Luthfi, E. T. (2009). *Algoritma Data Mining*. Yogyakarta. 2009
- Feldman R. & Sanger, J. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*
- Gorunescu, F. (2011). *Data Mining Concepts Models and Techniques*. Springer-Verlag Berlin Heidelberg.
- Maarif, A. A. (2015). *Penerapan Algoritma TF-IDF untuk Pencarian Karya Ilmiah*
- Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, Vol. 60 No.5 pp. 503–520.
- Sahat, H., & Simarankir, M. (2013). Studi Perbandingan Algoritma - Algoritma Stemming untuk Dokumen Teks Bahasa Indonesia. *Journal INFOKAR*, Vol. 1 No. 9 pp. 51–58.