

SISTEM PENILAIAN OTOMATIS JAWABAN ESAI PADA ELEARNING BELAJARDISINI.COM

Eko Sakti Pramukantoro¹

¹Fakultas Ilmu komputer Universitas Brawijaya
Email: ¹ekosakti@ub.ac.id

(Naskah masuk: 1 Oktober 2016, diterima untuk diterbitkan: 26 Desember 2016)

Abstrak

Belajardisini.com adalah sebuah *e-learning* berbasis *gamification*. Sistem tersebut menggunakan proses manual untuk melakukan koreksi jawaban uraian pendek(esai). Bertambahnya jumlah pelajar dan banyaknya ujian mengakibatkan pengajar harus meluangkan waktu untuk menilai jawaban soal ujian, sehingga penggunaan *e-learning* dan tanpa *e-learning* dari sisi efisiensi waktu tidak ada perbedaan. Oleh karena itu diperlukan sistem penilaian otomatis pada *e-learning* untuk jawaban esai. Sistem (*automated essay scoring*) AES dengan algoritma *Cosine Similarity* dikembangkan untuk menjawab kebutuhan tersebut. Algoritma cosine dipilih atas hasil penelitian sebelumnya yang membandingkan antara Cosine dan LSA, dimana Cosine lebih unggul dari LSA dari segi komputasi. Sistem yang diusulkan diintegrasikan ke dalam *e-learning*, sehingga perlu analisis performa sistem. Untuk mengetahui performa sistem dilakukan uji berdasarkan penggunaan CPU, Memori dan *page load time*. Proses uji performa *server* dari dua jenis soal yang diujikan disimpulkan bahwa semakin besar pengguna yang mengakses sistem maka semakin besar juga *CPU Usage* yang dibutuhkan yaitu paling besar adalah 0,2556 %. Untuk pengujian *page load time* dan *memory usage* tidak ditemukan perbedaan yang cukup signifikan ketika sistem digunakan oleh satu pengguna maupun banyak pengguna yaitu antara 0,208392 detik sampai 0,406842 detik untuk *page load time* dan antara 1,49 % sampai 1,56 % untuk *memory usage*.

Kata kunci: *automated essay scoring, cosine similarity, elearning*

Abstract

Belajardisini.com is elearning based on gamification concept. In that system for correction the answer of essay exam is still done manually. Increasing the number of students and the number of exam makes teachers having to take time to assess the answers to the exam, so using elearning and without elearning in assessment there is no different especially in term time efficiency. Therefore we need an automatic scoring system for the essay. Automated essay scoring system using cosine similarity is developed to answer those needs. Cosine algorithm is selected based on the results of previous studies that comparing cosine and LSA, where cosine superior in terms of computation. The proposed system is integrated into the e-learning, so analysis of system performance is required. Analysis system performance based on the use of CPU, memory and page load time. The result shown server performance of the two types of questions that tested concluded that the users accessing the system, the greater the required CPU Usage is at most 0.2556%. For the test page load time and memory usage found no significant difference when the system is used by a single user or multiple users is between 0.208392 seconds to 0.406842 seconds to page load time and between 1.49% to 1.56% for memory usage.

Keywords: *automated essay scoring, cosine similarity, elearning*

1. PENDAHULUAN

Dalam bidang pendidikan, situs web bisa digunakan sebagai sarana pembelajaran antara dosen dan mahasiswa yang disebut *elearning*. Pada penelitian sebelumnya dihasilkan sebuah media pembelajaran berbasis *gamification*, yaitu memadukan konsep *game* dalam sebuah *elearning* (Sakti, 2015). Dalam *elearning* tersebut tugas yang diberikan oleh dosen dapat dikerjakan oleh mahasiswa secara online. Tipe tugas yang ditawarkan dalam media belajar tersebut dapat dikategorikan menjadi dua jenis soal pilihan ganda dan soal uraian pendek atau esai.

Dalam perkembangannya fitur ujian esai menjadi kendala, karena bertambahnya jumlah pelajar dan banyaknya ujian mengakibatkan pengajar atau dosen harus meluangkan waktu untuk menilai jawaban soal ujian. Pada umumnya sistem koreksi jawaban ujian esai dilakukan secara manual, sehingga penggunaan *e-learning* dan tanpa *e-learning* dari sisi efisiensi waktu tidak ada perbedaan. Oleh karena itu dibutuhkan sistem penilaian otomatis pada ujian esai untuk mempercepat proses penilaian jawaban mahasiswa, akan tetapi tidak boleh mengesampingkan akurasi penilaian.

Ujian esai berbeda dengan ujian pilihan ganda yang hanya ada benar dan salah. Penilaian pada ujian esai lebih sulit karena harus menganalisis dan

mengerti teks yang ditulis mahasiswa. Ketika penilaian otomatis diterapkan pada *elearning* ada faktor yang tak kalah penting, yaitu performa server dengan sumber daya yang dibutuhkan. Perlu sebuah sistem penilaian otomatis dengan komputasi ringan. Hal ini yang menjadi kebutuhan khusus terhadap pengembangan sistem penilaian otomatis. Permasalahan tersebut dijawab dalam penelitian (Eko Sakti, 2016) dengan membandingkan algoritma *string similarity* (cosine) dan *corpus based* (*latent semantic analysis*). Dari penelitian tersebut didapat kesimpulan kedua algoritma mempunyai akurasi penilaian yang kompetitif akan tetapi cosine memiliki performa komputasi yang lebih baik dibandingkan dengan LSA, sehingga *cosine similarity* yang diusulkan untuk diterapkan pada AES.

Pada penelitian ini dilakukan integrasi sistem AES berdasarkan algoritma cosine similarity dan dilakukan pengujian untuk mengetahui performa sistem AES yang telah dibangun. Parameter uji performa sistem berdasarkan penggunaan CPU, memori dan page load. Sehingga dengan adanya sistem AES dapat membantu dosen atau pengajar dalam proses koreksi jawaban esai.

2. TEORI

2.1 Automated essay scoring

Automated essay scoring (AES) adalah sebuah sistem penilaian otomatis dengan cara membandingkan dua jawaban kemudian dikalkulasi hasil dari perbandingan menggunakan metode-metode yang ada. Contoh metode yang bisa digunakan untuk penerapan AES adalah *Corpus based similarity* dan *String based similarity*. *Corpus Based Similarity* (CBS) adalah metode yang mengukur kesamaan arti dari setiap kata berdasarkan sebuah kamus yang telah disediakan (Gomma, 2012). *Latent Semantic Analysis* (LSA) merupakan teknik yang populer dari CBS. *String Based Similarity* (CBS) adalah metode yang mengukur kesamaan *string* antara dua text tanpa melihat arti dari text tersebut. Teknik yang bisa digunakan untuk CBS adalah *Cosine Similarity*, *Dice* dan *Jaccard*.

2.2 Cosine Similarity

Cosine similarity merupakan metode fungsi pengukuran kesamaan antara dua vektor yang berbeda dengan mengukur cosinus dari sudut antara mereka (Kurniawan, 2014).

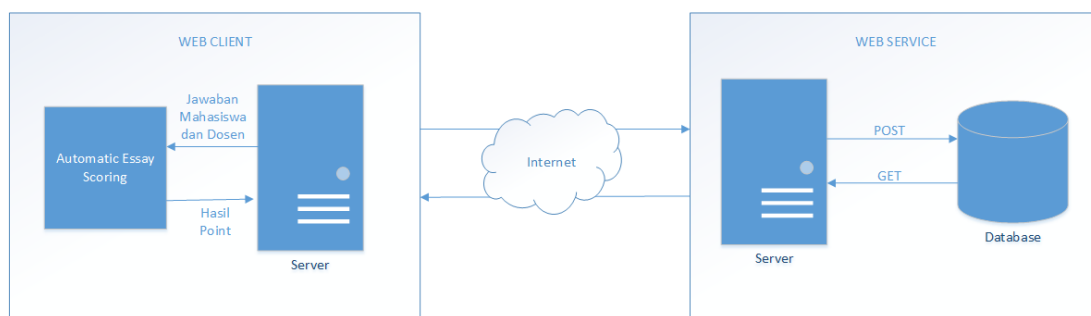
Metode ini dapat digunakan untuk melihat seberapa miripkah sebuah kalimat antara dua dokumen. Semakin besar nilai cosinus (maksimal 1) maka semakin mirip dokumen yang dibandingkan. Nilai cosinus 1 menyatakan kemiripan 100% sedangkan jika nilai 0 ketidakmiripannya 100%.

3. Sistem AES

Elearning yang telah dikembangkan berjalan diatas mesin dengan spesifikasi 1 GB *Memory*, 30 GB *Disk*, 2 TB *Transfer* dan dapat diakses melalui domain <http://belajardisini.com>. Aplikasi tersebut berbasis website dan dapat diakses oleh pengguna melalui sebuah aplikasi browser.

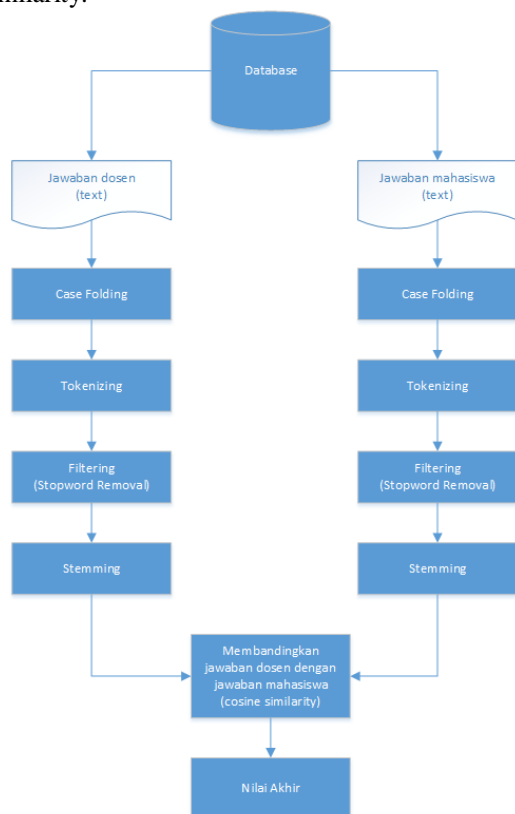
Untuk aplikasi sistem penilaian otomatis jawaban uraian pendek pada *elearning* ini dibagi atas dua sistem yaitu *web service* dan *web client*. Kedua sistem tersebut berkomunikasi dengan menggunakan *Web Service*. Dosen dapat melihat mata kuliah diajar dan dapat mengelola *challenge* uraian pendek yang nantinya bisa dikerjakan oleh mahasiswa. Semua proses data yang dilakukan pada *web client* nantinya akan dikirim ke *web service* dan disimpan ke *database*.

Gambar 1, menunjukan terdapat dua sistem utama yaitu *web client* dan *web service*. *Web service* berguna mengambil data jawaban dosen dan mahasiswa pada *database* kemudian dikirimkan ke *web client*. Setelah data jawaban diterima oleh *web client* kemudian akan masuk ke proses *Automatic Essay Scoring* (AES). Proses AES diletakkan pada *web client* dikarenakan proses AES hanya membutuhkan data jawaban yang diproses pada halaman detail jawaban mahasiswa. Data jawaban nantinya akan diproses pada AES dengan menggunakan metode *cosine similarity*. Jawaban dosen dan jawaban mahasiswa dibandingkan kemudian akan muncul hasil *point* berdasarkan perhitungan *cosine similarity*.



Gambar 1. Arsitektur komunikasi sistem AES

Proses pengoreksian otomatis pada dosen dapat dijabarkan lagi menjadi beberapa proses. Pertama jawaban dari dosen dan mahasiswa harus melalui text processing untuk menghilangkan kata-kata yang tidak memiliki arti penting kemudian baru masuk ke proses perhitungan dengan metode cosine similarity.



Gambar 2. Diagram Alur Sistem Penilaian Otomatis

Gambar 2 menjelaskan mengenai alur sistem pada penilaian otomatis jawaban uraian pendek. Jawaban yang sudah dosen inputkan langsung tersimpan di dalam *database*, selanjutnya jawaban tersebut akan diolah oleh sistem. Langkah yang pertama, jawaban tersebut akan mengalami proses *case folding*. Pada *case folding* ini variasi huruf harus diseragamkan dan tanda baca harus dihilangkan untuk menghilangkan *noise* pada saat pengambilan informasi. Selanjutnya langkah kedua, jawaban tersebut akan mengalami *tokenizing*. Proses *tokenizing* ini dilakukan untuk pemecahan kalimat menjadi kata.

Langkah yang ketiga yaitu melakukan *filtering* (*stopward removal*), proses ini dilakukan untuk pembuangan kata yang tidak penting. Dengan dibuangnya *stopwords*, ukuran kosakata menjadi berkurang sehingga hanya kata-kata penting yang terdapat dalam dokumen dan diharapkan akan menjadi kata-kata yang memiliki bobot yang tinggi. Langkah yang terakhir yaitu *stemming*, proses perubahan suatu kata menjadi kata dasar. Setelah melewati pengolahan jawaban seperti yang telah

dijelaskan diatas, maka jawaban akan dicocokkan dengan jawaban dari mahasiswa.

Jawaban mahasiswa sebelumnya juga mengalami proses yang sama seperti diatas sebelum akhirnya dicocokkan dengan jawaban dosen. Setelah melakukan pencocokan jawaban dosen dan mahasiswa, maka akan dihitung penilaiannya menggunakan Cosine Similarity. Setelah mendapatkan penilaian, data nilai akan disimpan di database dan juga hasil nilai tersebut akan keluar sebagai output.

Untuk proses perhitungan otomatisnya menggunakan metode *cosine similarity*. Script *cosine similarity* ditunjukkan pada Tabel 1

Tabel 1. Script cosine similarity

```

<?php

class CosineSimilarity {

public function similarity(array $vec1,
array $vec2) {
    return $this->_dotProduct($vec1,
$vec2) / ($this->_absVector($vec1) *
$this->_absVector($vec2));
}

public function _dotProduct(array
$vec1, array $vec2) {
    $result = 0;
    foreach (array_keys($vec1) as $key1) {
        foreach (array_keys($vec2) as $key2) {
            if ($key1 === $key2) $result +=
$vec1[$key1] * $vec2[$key2];
        }
    }
    return $result;
}

public function _absVector(array $vec)
{
    $result = 0;
    foreach (array_values($vec) as $value)
    {
        $result += $value * $value;
    }
    return sqrt($result);
}
}
  
```

4. Pengujian dan pembahasan

Pengujian dilakukan untuk mengetahui performa sistem AES yang telah diintegrasikan dengan aplikasi elearning. Parameter performa dengan melihat seberapa besar *Page load time*, *memory usage* dan *CPU Usage* yang digunakan oleh sistem AES.

Proses pengujian performa server berfungsi untuk melihat seberapa besar pengaruh proses penilaian otomatis terhadap performa server. Proses pengujian performa server dibantu menggunakan *Webserver Stress Tool* yang berfungsi untuk membuat skenario seberapa banyak pengguna yang

mengakses sistem. Terdapat tiga skenario yang digunakan. Skenario pertama hanya ada satu

Tabel 2. Hasil Pengujian *Page Load Time*

JENIS SOAL	Skenario 1	Skenario 2	Skenario 3
Soal A	0,231384	0,228456	0,406842
Soal B	0,234756	0,208392	0,209024

Tabel 3. Hasil Pengujian *Memory Usage*

JENIS SOAL	Skenario 1	Skenario 2	Skenario 3
Soal A	1,5	1,49	1,5
Soal B	1,56	1,5	1,5

Tabel 4. Hasil Pengujian *CPU Usage*

JENIS SOAL	Skenario 1	Skenario 2	Skenario 3
Soal A	0,011	0,1822	0,2556
Soal B	0,0308	0,1484	0,2318

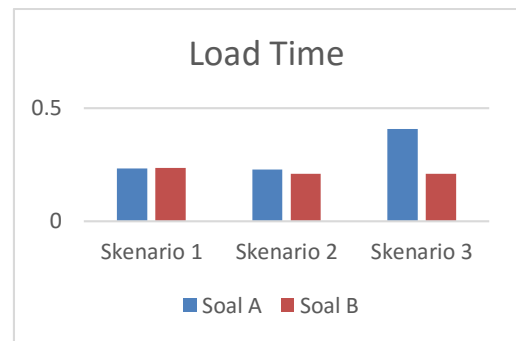
pengguna yang mengakses sistem, skenario kedua 50 pengguna kemudian skenario ketiga 100 pengguna.

Proses pengujian dilakukan terhadap dua jenis soal uraian pendek yaitu soal sebutkan dan apa perbedaan. Performa server yang dihitung adalah *page load time*, *memory usage* dan *CPU usage*. Pengujian dilakukan terhadap 50 jawaban mahasiswa dari masing-masing jenis soal. Dari hasil pengujian yang dilampirkan pada lampiran diambil hasil rata-rata nya kemudian dipisahkan antara hasil *page load time*, *memory usage* dan *CPU usage*.

Berdasarkan kasus uji pada Tabel 2, didapatkan hasil sebagai berikut :

- Pada skenario 1 rata-rata *Page Load Time* yang dibutuhkan untuk mengakses jenis soal A adalah 0,231384 detik dan jenis soal B adalah 0,234756 detik.
- Pada skenario 2 rata-rata *Page Load Time* yang dibutuhkan untuk mengakses jenis soal A adalah 0,228456 detik dan jenis soal B adalah 0,208392 detik.
- Pada skenario 3 rata-rata *Page Load Time* yang dibutuhkan untuk mengakses jenis soal A adalah 0,406842 detik dan jenis soal B adalah 0,209024 detik.

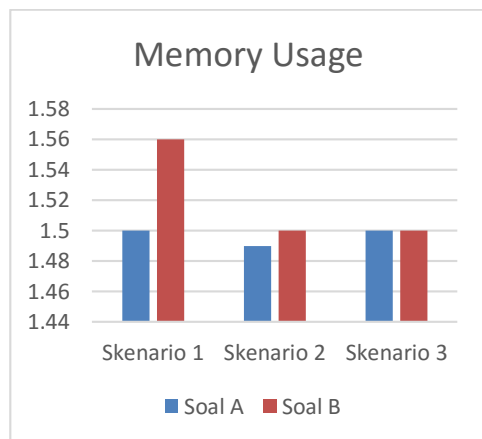
Pada Gambar 3 merupakan grafik dari hasil pengujian *page load time*. Berdasarkan grafik untuk hasil pengujian antara skenario 1 dan skenario 2 tidak mempunyai perbedaan yang signifikan. Untuk skenario 3 perbedaan yang besar terdapat pada pengujian jenis soal A yang membutuhkan waktu lebih banyak dari pada skenario lainnya



Gambar 3. Grafik Hasil Pengujian *Page Load Time*

Berdasarkan kasus uji pada Tabel 3 didapatkan hasil sebagai berikut :

- Pada skenario 1 rata-rata *Memory Usage* yang dibutuhkan untuk mengakses jenis soal A adalah 1,5 MB dan jenis soal B adalah 1,56 MB.
- Pada skenario 2 rata-rata *Memory Usage* yang dibutuhkan untuk mengakses jenis soal A adalah 1,49 MB dan jenis soal B adalah 1,5 MB.
- Pada skenario 3 rata-rata *Memory Usage* yang dibutuhkan untuk mengakses jenis soal A adalah 1,5 MB dan jenis soal B adalah 1,5 MB.

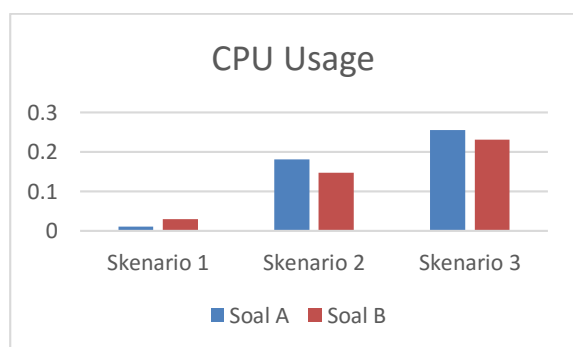


Gambar 4. Grafik Hasil Pengujian *Memory Usage*

Pada Gambar 4 merupakan grafik dari hasil pengujian *memory usage*. Tidak ada perbedaan yang signifikan antara ketiga skenario kecuali pada skenario 1 untuk jenis soal B yang membutuhkan *memory usage* sedikit lebih besar daripada lainnya.

Berdasarkan kasus uji pada tabel 4 didapatkan hasil sebagai berikut :

- Pada skenario 1 rata-rata *CPU Usage* yang dibutuhkan untuk mengakses jenis soal A adalah 0,011 % dan jenis soal B adalah 0,0308 %.
- Pada skenario 2 rata-rata *CPU Usage* yang dibutuhkan untuk mengakses jenis soal A adalah 0,1822 % dan jenis soal B adalah 0,1484 %.
- Pada skenario 3 rata-rata *CPU Usage* yang dibutuhkan untuk mengakses jenis soal A adalah 0,2556 % dan jenis soal B adalah 0,2318 %.



Gambar 5. Grafik Hasil Pengujian *CPU Usage*

Pada Gambar 5 merupakan grafik hasil pengujian *CPU Usage*. Terdapat perbedaan yang cukup signifikan antara ketiga skenario. Semakin banyak pengguna yang mengakses sistem maka semakin besar pula *CPU Usage* yang dibutuhkan.

5. Kesimpulan

Berdasarkan hasil pengujian terhadap performa *server* dapat disimpulkan dari dua jenis soal yang diujikan bahwa semakin besar pengguna yang mengakses aplikasi maka semakin besar juga *CPU Usage* yang dibutuhkan yaitu paling besar adalah 0,2556 %. Untuk pengujian *page load time* dan *memory usage* tidak ditemukan perbedaan yang cukup signifikan ketika sistem digunakan oleh satu pengguna maupun banyak pengguna yaitu antara 0,208392 detik sampai 0,406842 detik untuk *page load time* dan antara 1,49 % sampai 1,56 % untuk *memory usage*.

Daftar Pustaka

- Afandi, A. (2012). Implementasi Algoritma Jaro Winkler Distance Untuk Aplikasi Penilaian Otomatis.
- Eko Sakti, M. A. (2016). Comparative Analysis Of String Similarity And Corpus-Based Similarity For Automatic Essay Scoring System On E-Learning Gamification.
- Fuat, R. (2011). Sistem Penilaian Uraian Pendek Otomatis pada E-Learning dengan Metode Cosine Similarity.
- Gomma, W. (2012). Short Answer Grading Using Similarity and Corpus Based Similarity.
- Kurniawan, A. (2014). Perancangan dan Pembuatan Aplikasi Pencarian Informasi Beasiswa dengan Menggunakan Cosine Similarity.
- Sakti, E. (2015). Elearning Berbasis Gamification.
- Thada, V. (2013). *Comparison of Jaccard, Dice, Cosine Similarity*.