

APLIKASI SISTEM AUTOMATED ESSAY SCORING UNTUK JAWABAN SOAL UJIAN DENGAN MENERAPKAN ALGORITMA JARO WINKLER

Krisnawati Harefa¹, Abdul Jabbar²

^{1,2}Program Studi Teknik Informatika,Fakultas Teknik dan Komputer, Universitas Harapan
Medan

Email : ¹sinaharefa1994@gmail.com, ² abduljabbararlbs@gmail.com

ABSTRAK

Pada perkembangannya fitur ujian esai menjadi kendala, karena bertambahnya jumlah siswa dan banyaknya ujian mengakibatkan pengajar 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. Automated essay scoring adalah sebuah sistem penilaian otomatis dengan cara membandingkan dua jawaban kemudian dikalkulasi hasil dari perbandingan menggunakan algoritma yang ada. Salah satu algoritma yang digunakan adalah algoritma Jaro-Winkler Distance, yaitu untuk mengukur kesamaan antara dua string. Pada penelitian ini dilakukan integrasi sistem automated essay scoring berdasarkan algoritma Jaro-Winkler Distance, sehingga dengan adanya sistem automated essay scoring dapat membantu pengajar dalam proses koreksi jawaban esai. Hasil pengujian sistem menunjukkan, penilaian jawaban yang dihasilkan sistem dipengaruhi oleh ada tidaknya term kunci pada kalimat jawaban. Semakin banyak perbedaan jumlah kata pada suatu kalimat antara guru dan siswa, maka nilai yang dihasilkan semakin kecil. Jika suatu kalimat jawaban siswa memiliki persebaran term kunci yang semakin sama dengan kunci jawaban guru maka, nilai yang dihasilkan juga semakin tinggi.

Kata Kunci : *Sistem Penilaian Otomatis, Essay, Algoritma Jaro Winkler*

ABSTRACT

In its development, the essay exam features become an obstacle, because the increase in the number of students and the number of examinations results in the teacher taking the time to assess the answers to the exam questions. In general, the answer correction system for essay answers is done manually, so the use of e-learning and without e-learning in terms of time efficiency there is no difference. Therefore, an automatic assessment system is needed on essay examinations to accelerate the process of evaluating student answers. Automated essay scoring is an automatic scoring system by comparing two answers and then calculating the results of a comparison using an existing algorithm. One algorithm used is the Jaro-Winkler Distance algorithm, which is to measure the similarity between two strings. In this research, an automated essay scoring system integration is based on the Jaro-Winkler Distance algorithm, so that with an automated scoring essay system, it can help the teacher in the essay answer correction process. The results of the system testing show, the assessment of the answers generated by the system is influenced by whether or not there is a key term in the answer sentence. The more different the number of words in a sentence between teacher and student, the smaller the resulting value. If a student's answer

sentence has a key term distribution which is the same as the teacher's answer key then the resulting value is also higher.

Key Word : Automated Essay Scoring, Essay, Algoritma Jaro Winkler

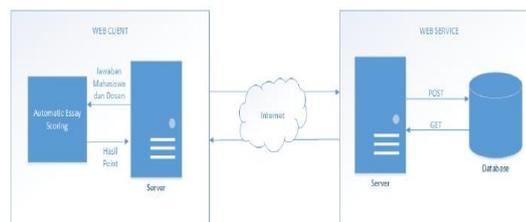
1. PENDAHULUAN

Dalam bidang pendidikan, situs *web* bisa digunakan sebagai sarana pembelajaran antara dosen dan mahasiswa yang disebut *e-learning*. Dalam *e-learning* tersebut tugas yang diberikan oleh dosen dapat dikerjakan oleh mahasiswa secara *online*. Setiap proses pembelajaran memerlukan suatu evaluasi berupa ujian, begitu pula dengan *e-learning*. Pada proses *e-learning* jenis ujian yang banyak digunakan adalah jenis ujian pilihan ganda dan isian singkat. Alasannya adalah kemudahan dalam proses penilaian, komputer yang menjadi komponen penting dalam proses *e-learning* lebih mudah dalam melakukan penilaian ujian pilihan ganda dan isian singkat secara akurat dibandingkan dengan melakukan penilaian jenis ujian esai.

Dalam perkembangannya fitur ujian esai menjadi kendala, karena bertambahnya jumlah siswa dan banyaknya ujian mengakibatkan pengajar meluangkan waktu untuk menilai jawaban soal ujian. Oleh karena itu dibutuhkan sistem penilaian otomatis pada ujian esai untuk mempercepat proses penilaian jawaban mahasiswa, akan tetapi tidak boleh mengesampingkan akurasi penilaian. *Automated essay scoring* adalah sebuah sistem penilaian otomatis dengan cara membandingkan dua jawaban kemudian dikalkulasi hasil dari perbandingan menggunakan algoritma yang ada. Salah satu algoritma yang digunakan adalah algoritma *Jaro-Winkler Distance*, yaitu untuk mengukur kesamaan antara dua *string*. Penggunaan algoritma *Jaro Winkler* dalam aplikasi ini adalah untuk mencocokkan antara jawaban dengan kunci jawaban, yaitu untuk membandingkan kecocokan tingkat kesamaan kalimat.

E-learning merupakan proses belajar secara efektif yang dihasilkan dengan cara menggabungkan penyampaian materi secara digital yang terdiri dari dukungan dan layanan dalam belajar [1]. Berdasarkan definisi tersebut, dapat disimpulkan bahwa *e-learning* adalah sebuah sistem pembelajaran yang menggunakan komputer dan internet sebagai medianya.

Automated Essay Scoring (AES) merupakan salah satu cabang dari bidang *document classification* [2]. AES merupakan teknologi yang sedang berkembang, yang digunakan untuk menilai esai secara otomatis. AES adalah sebuah sistem penilaian otomatis dengan cara membandingkan dua jawaban kemudian dikalkulasi hasil dari perbandingan menggunakan metode-metode yang ada [3].



Gambar 1 Arsitektur Komunikasi Sistem AES

Gambar 1 menunjukkan 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 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 algoritma *Jaro Winkler*. Jawaban dosen

dan jawaban mahasiswa dibandingkan kemudian akan muncul hasil *point* berdasarkan perhitungan algoritma *Jaro Winkler*.

Text mining memiliki definisi menambang data yang berupa teks. *Text mining* merupakan penerapan konsep dan teknik data mining untuk mencari pola dalam teks, yaitu proses menganalisis teks guna menyarikan informasi yang bermanfaat untuk tujuan tertentu. Berdasarkan ketidakteraturan struktur data teks, maka proses *text mining* memerlukan beberapa tahap awal yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur [4]. *Text mining* merupakan salah satu bidang khusus dari *data mining*. *Text mining* lebih berfokus pada teknik dan metodologi dalam lingkup temu kembali sistem informasi. *Text mining* berfungsi untuk mendapatkan informasi dari sebuah dokumen teks dalam penggalian informasi terdapat beberapa tahapan yang diantaranya adalah [5] :

1. *Text preprocessing*

Text preprocessing merupakan tahap awal dari *text mining* dimana data *text* akan dibersihkan sehingga *text* menjadi lebih terstruktur sebelum masuk ketahap berikutnya untuk diolah lebih lanjut. Sekumpulan karakter yang bersambungan (teks) harus dipecah-pecah menjadi lebih berarti. Hal tersebut dapat dilakukan dalam beberapa tingkatan yang berbeda. Suatu dokumen dapat di pecah menjadi suatu bab, paragraf, kalimat dan kata. Tahapan *text preprocessing* meliputi:

a. *Case Folding*

Tahap *case folding* merupakan tahap dimana semua huruf dalam dokumen teks diubah menjadi huruf kecil dan hanya huruf a sampai z saja yang diterima.

b. *Filtering*

Pada tahap *filtering* dimana akan dilakukan penghapusan *stopwords* untuk mengeliminasi kata-kata yang tidak deskriptif dan yang sering muncul.

c. Penghapusan spasi

Tahap ini merupakan tahap dimana semua spasi yang terdapat pada dokumen teks dihilangkan.

2. *Pattern Discovery*

Pattern discovery merupakan tahap dari terakhir dari *text mining* dimana dokumen teks yang telah dibersihkan dan sudah terstruktur akan diproses untuk mendapatkan pola dan informasi dari dokumen teks tersebut.

Jaro Winkler Distance memiliki kelebihan yaitu dapat mengukur kesamaan antara dua *string* dimana metode ini biasanya digunakan di dalam pendeteksian duplikat dokumen. Semakin tinggi nilai *Jaro-Winkler Distance* untuk dua buah *string*, maka semakin mirip *string* tersebut. Nilai normalnya adalah 0 yang menandakan tidak adanya kesamaan dan 1 yang menandakan adanya kesamaan [6].

Langkah dasar untuk menghitung algoritma *Jaro Winkler* antara dua *string* S1 dan S2 adalah sebagai berikut [9] :

1. Menghitung panjang *string* S1 dan S2
2. Menemukan jumlah karakter yang “sama persis”(m) dalam dua string yang dibandingkan
3. Menghitung jumlah transposisi kedua buah *string*

Pada algoritma *Jaro-Winkler Distance* digunakan rumus untuk menghitung jarak (dj) antara dua *string* yaitu S1 dan S2 adalah :

$$d_j = \frac{1}{3} \times \left(\frac{m}{S_1} + \frac{m}{S_2} + \frac{m - t}{m} \right)$$

Dimana :

m = Jumlah karakter yang sama persis

|S₁| = Panjang *string* 1

|S₂| = Panjang *string* 2

t = Jumlah transposisi

Jarak teoritis dua buah karakter yang disamakan dapat dibenarkan jika tidak melebihi :

$$\left(\frac{\max(|S_1|, |S_2|)}{2} \right) - 1$$

Akan tetapi bila mengacu kepada nilai yang akan dihasilkan oleh algoritma *Jaro-Winkler Distance* maka nilai maksimalnya adalah 1, yang menandakan kesamaan *string* yang dibandingkan mencapai seratus persen atau sama persis. Algoritma *Jaro-Winkler Distance* menggunakan *prefix scale* (p), menurut *Winkler* nilai standar untuk konstanta ini adalah $p = 0.1$. Dan *prefix length* (l) yaitu untuk menyatakan panjang *prefix* atau panjang karakter yang sama sampai ditemukan ketidaksamaan (maksimum 4 karakter), maka untuk menghitung *Jaro-Winkler Distance* (d_w) adalah :

$$d_w = d_j + (l * p(1 - d_j))$$

Dimana :

d_j = *Jaro Distance* untuk *string* S_1 dan S_2

l = Panjang *prefix* (panjang karakter yang sama sebelum ditemukan ketidaksamaan) nilai maksimum 4 karakter

p = Konstanta *scaling* faktor (nilai standar untuk konstanta ini menurut *Winkler* adalah $p = 0.1$).

Unified Modelling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem [10].

UML sendiri terdiri atas pengelompokan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. UML mempunyai beberapa sub diagram diantaranya yaitu *use-case diagram* dan *activity diagram*.

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut [7].

Flowchart adalah cara penulisan algoritma dengan menggunakan notasi grafis. *Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan antara proses digambarkan dengan garis penghubung [8]. Tujuan penggunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi, dan jelas dengan menggunakan simbol-simbol yang standar. Tahap penyelesaian masalah yang disajikan harus jelas, sederhana, efektif dan tepat

Database adalah kumpulan dari *item* data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, yang kelak dapat dimanfaatkan kembali dengan cepat dan mudah [9].

MySQL (*My Structure Query Language*) adalah sistem manajemen *database* relasi (relation data-base Management system) yang bersifat “terbuka” (*open Source*). Terbuka maksudnya adalah *MySQL* dapat digunakan oleh siapa saja, baik versi kode program aslinya maupun versi binernya (*executable* program) dan bisa digunakan secara gratis baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi komputer [10].

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan

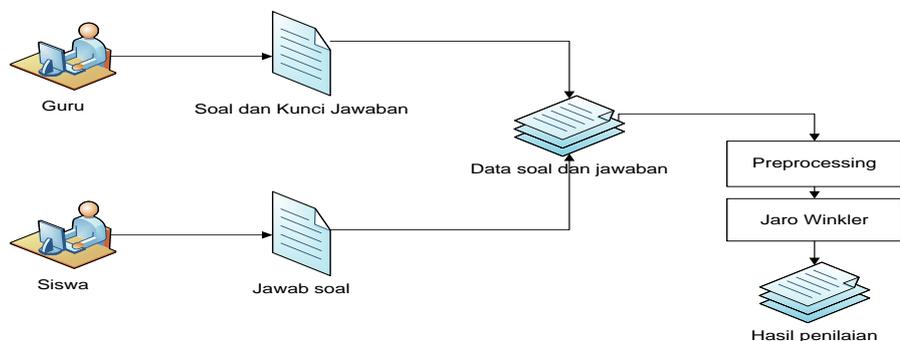
dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti *C*, *Java*, dan *Perl* serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server-side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, serverlah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan [11].

2. METODE PENELITIAN

Penelitian ini pada dasarnya mengedepankan aspek implementasi dan analisis suatu teori, tentang penilaian esai secara otomatis dengan menggunakan pendekatan algoritma *Jaro Winkler*. Seperti pada penelitian-penelitian tentang analisis atau pembuktian suatu teori yang banyak menerapkan metode studi kepustakaan, begitu juga dalam penelitian ini,

Input dari penelitian ini adalah kumpulan dari jawaban *essay* siswa. Output dari penelitian ini adalah berupa hasil *scoring* penilaian jawaban siswa yang dibandingkan dengan jawaban dari guru, dimana prosesnya dengan menggunakan algoritma *Jaro Winkler* untuk menentukan tingkat kemiripan antara jawaban siswa dengan jawaban guru.

Gambar 2.1 dibawah ini memperlihatkan arsitektur sistem penilaian otomatis jawaban siswa dengan menggunakan algoritma *Jaro Winkler*.



Gambar 2 Arsitektur Sistem Penilaian Otomatis Jawaban Siswa

Gambar 2 menunjukkan bahwa proses inputan dilakukan oleh guru dan siswa, selanjutnya dilakukan proses *preprocessing* yang meliputi proses tokenisasi dimana pada proses ini semua kalimat akan dijadikan token kata kemudian dilakukan proses menghapus stopword yang membuang semua kata umum yang tidak memiliki makna, proses stemming yang mengubah menjadi kata dasar. Pada tahap akhir dilakukan proses hasil penilaian yang diukur menggunakan algoritma *Jaro Winkler*.

Untuk mempermudah menganalisis sebuah sistem dibutuhkan dua jenis kebutuhan yaitu kebutuhan non fungsional dan kebutuhan fungsional. Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Sedangkan kebutuhan non fungsional adalah kebutuhan yang menitik beratkan pada properti perilaku yang dimiliki oleh sistem. Tujuan dari analisis kebutuhan sistem adalah memahami dengan sesungguhnya kebutuhan dari sistem.

Algoritma *Jaro Winkler* merupakan varian dari *Jaro Distance Metric* yaitu sebuah algoritma untuk mengukur kesamaan antara dua *string*, biasanya algoritma ini digunakan di dalam pendeteksian duplikat. Semakin tinggi *Jaro Winkler Distance* untuk dua *string* maka semakin mirip dengan *string* tersebut. Nilai normalnya ialah nol menandakan tidak ada kesamaan dan dkk yang menandakan adanya kesamaan. Dasar dari algoritma ini memiliki tiga bagian yaitu sebagai berikut :

1. Menghitung panjang *string*
2. Menemukan jumlah karakter yang sama di dalam dua *string*
3. Menemukan jumlah transposisi

Pada algoritma *Jaro Winkler* untuk menghitung jarak (d_j) antara dua *string* yaitu S_1 dan S_2 digunakan persamaan berikut :

$$d_j = \frac{1}{3} \left(\frac{c}{|S_1|} + \frac{c}{|S_2|} + \frac{c-t}{c} \right)$$

Dimana c merupakan jumlah karakter yang sama persis, $|S_1|$ merupakan panjang *string* 1 dan $|S_2|$ merupakan panjang *string* 2, serta t merupakan jumlah transposisi. Jarak teoritis dua buah karakter yang dianggap sama dapat dikatakan benar jika tidak melebihi batas seperti yang tercantum pada persamaan berikut:

$$\text{jarak teoritsi} = \left(\frac{\max(|S_1|, |S_2|)}{2} \right) - 1$$

Jaro Winkler Distance menggunakan *prefix scale* (p) yang memberikan tingkat penilaian yang lebih dan *prefix length* (ℓ) yang menyatakan panjang awalan yaitu panjang karakter yang sama dengan *string* yang dibandingkan sampai ditemukan ketidaksamaan, jumlah karakter yang sama sebelum ditemukan ketidaksamaan adalah maksimal 4. Bila *string* dan yang dibandingkan, maka *Jaro Winkler Distance*-nya ditentukan dengan persamaan berikut ini:

$$d_j = \frac{1}{3} \left(\frac{c}{|S_1|} + \frac{c}{|S_2|} + \frac{c-t}{c} \right)$$

Dimana d_j merupakan *Jaro Distance* untuk *string* yaitu S_1 dan S_2 , ℓ merupakan panjang *prefiks* umum di awal *string* (panjang karakter yang sama sebelum ditemukan ketidaksamaan max 4) serta p merupakan konstanta *scaling factor* ($p=0,1$).

Diberikan contoh kata yang salah yaitu “kosnultasi” dengan kata yang benar yaitu “konsultasi”. Berikut perhitungan manual nilai *distance*-nya yaitu sebagai berikut.

$$S_1 = \text{kosnultasi}$$

$$S_2 = \text{konsultasi}$$

Berdasarkan kata diatas maka dapat diketahui bahwa :

$c = 10$, yaitu merupakan jumlah karakter yang sama persis

$S_1 = 10$, yaitu merupakan panjang *string* 1

$S_2 = 10$, yaitu merupakan panjang *string* 2

Selanjutnya karena karakter yang tertukar hanyalah “s” dan “n” maka dapat diketahui jumlah transposisi $t = 1$, sehingga nilai *Jaro distance*-nya adalah:

$$d_j = \frac{1}{3} \left(\frac{c}{|S_1|} + \frac{c}{|S_2|} + \frac{c-t}{c} \right)$$

$$d_j = \frac{1}{3} \left(\frac{10}{|10|} + \frac{10}{|10|} + \frac{10-1}{10} \right)$$

$$d_j = 0,967$$

Bila diperhatikan susunan S_1 dan S_2 dapat diketahui nilai *prefix length* (ℓ) = 2 karena karakter yang sama sebelum ditemukan kesalahan adalah karakter ‘k’ dan karakter ‘o’. Nilai konstanta *scaling factor* $p = 0,1$. Maka nilai *Jaro Winkler Distance* adalah:

$$dw = d_j + (\ell p(1 - d_j))$$

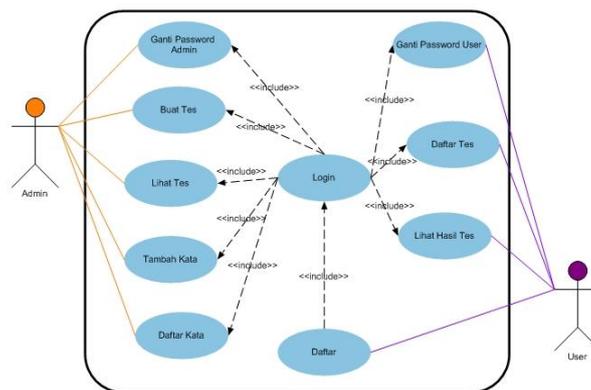
$$dw = 0,967 + (2 * 0,1(1 - 0,967))$$

$$dw = 0,9736$$

Oleh karena itu nilai *distance* untuk *Jaro Winkler* pada kata yang salah yaitu kosnultasi dan kata kamus yang menjadi target yaitu konsultasi adalah 0,9736.

Pada pemodelan sistem disini penulis membuat skenario sistem dalam bentuk diagram menggunakan *Unified Modeling Language (UML)*. Jenis pemodelan sistem yang digunakan yaitu *use case diagram* dan *activity diagram*. Sedangkan bentuk pemodelan sistem lainnya yang digunakan yaitu *flowchart* sistem dan perancangan *database*.

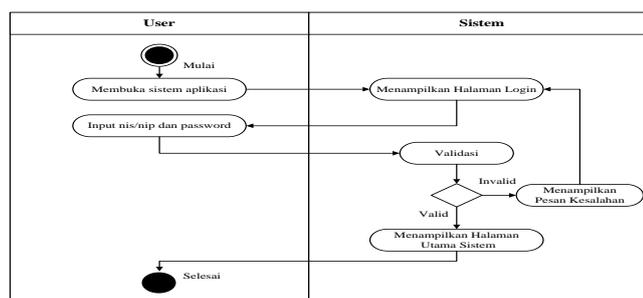
Use case diagram digunakan untuk mengetahui apa saja yang dapat dilakukan oleh pengguna (aktor) terhadap fungsionalitas yang terdapat pada aplikasi yang dibangun. *Use case diagram* pada aplikasi seperti terlihat pada gambar 2.2.



Gambar 3 Use Case Diagram Sistem

Berdasarkan gambar 3 terdapat dua aktor yang terlibat dalam sistem, yaitu *admin* dan *user*. Pada aktor *admin* terdapat lima *use case* yang dapat diakses, yaitu *use case* ganti password admin, buat tes, lihat tes, tambah kata, dan daftar kata, dimana keliam *use case* ini dapat diakses setelah melalui login terlebih dahulu. Sedangkan pada aktor *user* terdapat tiga buah *use case*, yaitu *use case* ganti password user, daftar tes, lihat hasil tes, dan daftar, dimana semua *use case* dapat diakses setelah melalui login kedalam sistem.

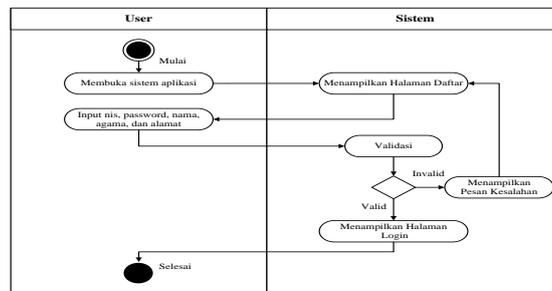
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Diagram ini sangat mirip dengan *flowchart* karena kita dapat memodelkan proses logika, proses bisnis dan alur kerja. Perbedaan utamanya adalah *flowchart* dibuat untuk menggambarkan alur kerja dari sebuah sistem, sedangkan *activity diagram* dibuat untuk menggambarkan aktivitas aktor. Adapun *activity diagram* sistem dalam proses login dapat digambarkan sebagai berikut.



Gambar 4 Activity Diagram Login

Berdasarkan gambar 4 yang harus dilakukan oleh *user* dalam *activity diagram login* adalah memasukkan *nis/nip* dan *password*, kemudian sistem akan memverifikasi data dengan melakukan pengecekan kedalam *database*. Jika datanya tidak benar (*invalid*) maka sistem akan menampilkan pesan kesalahan dan kembali ke halaman *login* untuk memasukkan *nis/nip* dan *password* yang benar. Jika *username* dan *password* yang dimasukkan benar, maka sistem akan menampilkan halaman utama.

Jika seorang *user* belum memiliki account untuk *login* maka harus mendaftar terlebih dahulu dengan mengakses *form* daftar *Activity diagram* dalam proses pendaftaran *user* dapat dilihat pada gambar 5



Gambar 5 Activity Diagram Daftar

Flowchart sistem bertujuan untuk memperlihatkan urutan atau langkah-langkah program dan hubungan antar proses beserta pernyataannya.

Perancangan sistem *database* ini merupakan rancangan sistem yang mengintegrasikan kumpulan data yang saling berhubungan satu dengan yang lainnya. *Database* ini didapat dari pemetaan (*mapping*) *class entity* yang telah digambarkan dalam *class diagram* sebelumnya.

Perancangan *layout* sistem merupakan tahapan yang dilakukan untuk mendesain tampilan halaman sistem. Secara umum, perancangan *layout* pada aplikasi sistem ini mencakup perancangan *layout* halaman *home*, perancangan *layout* halaman *admin* dan perancangan *layout* halaman *user*. Perancangan *layout* ini digunakan untuk tahapan sebelum melakukan proses *coding* program, yang bertujuan untuk mengetahui fungsi dari setiap halaman yang akan ditampilkan nantinya kepada *user*.

3. HASIL DAN PEMBAHASAN

Pada tahap penelitian ini dilakukan adalah membangun aplikasi dengan mengimplementasikan hasil dari tahapan desain interface kedalam bahasa pemrograman yang dipakai. Masukan pada tahapan ini yaitu data soal essay, sedangkan hasil atau output berupa hasil penilaian otomatis jawaban. Untuk tahapan ini sumber daya yang digunakan yaitu laptop atau komputer, bahasa pemrograman PHP dan MySQL.

Pada tahap ini dilakukan penerapan hasil perancangan antarmuka (*interface*) ke dalam sistem yang dibangun dengan menggunakan perangkat lunak yang telah dipaparkan pada sub bab implementasi perangkat lunak. Implementasi tahapan antarmuka (*interface*) pemakai merupakan bagian yang menyediakan sarana untuk user agar bisa berkomunikasi dengan sistem dalam bentuk program aplikasi. Hasil implementasi sistem merupakan beberapa contoh form atau halaman yang ada pada sistem, halaman ini tampil ketika user memanfaatkan fasilitas yang tersedia pada sistem.

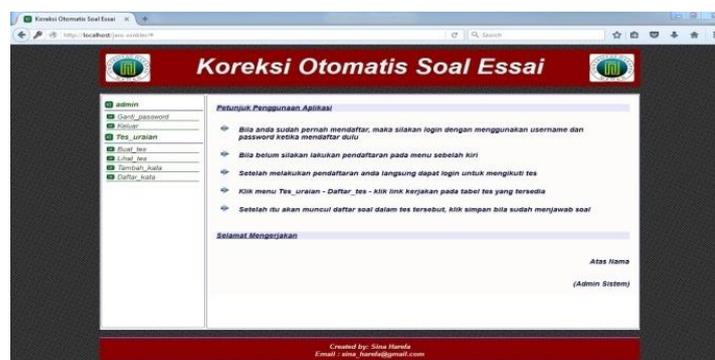
Database merupakan sistem yang terdiri atas kumpulan tabel data yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer. *Database* yang digunakan dalam pembuatan sistem ini adalah MySQL yang diakses menggunakan phpMyAdmin.

Halaman home merupakan halaman yang pertama kali ditampilkan saat user mengakses sistem. Form login ini difungsikan untuk mengakses masuk kedalam halaman utama sistem. Setelah user berhasil masuk kedalam sistem dengan menginputkan username dan password maka sistem akan menampilkan halaman utama user. Adapun implementasi hasil pengujian dari halaman home user dapat dilihat pada gambar 6 dibawah ini.



Gambar 6 Hasil Pengujian Halaman Home User

Setelah admin login kedalam sistem maka akan ditampilkan halaman utama admin. seperti terlihat pada gambar 7 berikut.



Gambar 7 Hasil Pengujian Halaman Admin

Gambar 7 menunjukkan hasil pengujian halaman admin, terdapat lima buah menu yaitu menu ganti password, menu keluar untuk log out dari sistem, menu buat tes, menu lihat tes, menu tambah kata, dan menu daftar kata.

Sedangkan jika admin memilih menu buat tes maka sistem membuat tes uraian, pilih nama pelajaran lalu inputkan nama tes, tanggal tes dan waktu tes selanjutnya pilih tombol simpan untuk menyimpan data kedalam database sistem. Jika ingin mengedit data tes maka pilih link edit pada baris tabel mata pelajaran yang diinginkan dan data tes yaitu dengan melakukan perubahan data yang diinginkan lalu pilih tombol simpan untuk mengupdate data kedalam database sistem. Sedangkan jika ingin menghapus data tes maka pilih link hapus pada baris tabel yang diinginkan dan sistem akan menghapus data tersebut. Sedangkan untuk menambah data soal maka pilih tombol tambah soal dan sistem akan menampilkan, soal tes uraian yang dilakukan dengan menginputkan pertanyaan atau soal kemudian menginputkan kunci jawaban, selanjutnya pilih tombol proses untuk mengatur bobot nilai jawaban yang diinginkan yang nantinya digunakan pada saat koreksi jawaban user. Sedangkan untuk melihat hasil jawaban user yang telah mengerjakan soal tes uraian dapat dilakukan dengan memilih menu lihat tes, kemudian pilih lihat hasil pada baris tabel yang diinginkan dan sistem akan menampilkan hasilnya. Sedangkan untuk

menambahkan kata kamus baru dapat dilakukan dengan memilih menu tambah kata dan sistem akan menampilkan halaman untuk membuat kamus baru dilakukan dengan menginputkan kata baru dan sinonim, selanjutnya pilih tombol simpan untuk menyimpan data kedalam database sistem. Untuk menampilkan daftar kamus kata dapat dilakukan dengan memilih menu daftar kata dan sistem akan menampilkan halaman daftar kamus.

Sedangkan untuk melihat hasil jawaban user yang telah mengerjakan soal tes uraian dapat dilakukan dengan memilih menu lihat tes, kemudian pilih lihat hasil pada baris tabel yang diinginkan dan sistem akan menampilkan hasilnya seperti terlihat pada gambar 8 dibawah ini.



The screenshot shows a web browser window with the title 'Koreksi Otomatis Soal Essai'. The main content area displays a table titled 'Daftar Nilai Tes Yang Telah Dikerjakan'. The table has columns for 'NO', 'ID_TES', 'ID_USER', 'MATA PELAJARAN', 'NAMA TES', 'NAMA SISWA', 'NILAI', and 'AKSI'. There are two rows of data in the table.

| NO | ID_TES | ID_USER | MATA PELAJARAN | NAMA TES | NAMA SISWA | NILAI | AKSI |
|----|--------|---------|----------------|---------------|-------------|-------|---------------|
| 1 | 1 | 133503 | Sesuai | LEAH SEMESTER | Sina Harita | 22.00 | Ditak.jawaban |
| 2 | 1 | 133501 | Sesuai | LEAH SEMESTER | Sina Harita | 12.00 | Ditak.jawaban |

Gambar 8 Hasil Pengujian Lihat Hasil Nilai Tes Uraian User

Pada gambar 8 menunjukkan hasil pengujian untuk menampilkan daftar nilai user yang telah mengikuti tes uraian berdasarkan mata pelajaran yang diikuti

4. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan, dapat diperoleh disimpulkan sebagai berikut:

1. Algoritma *Jaro Winkler* dapat diimplementasikan untuk membantu mengoreksi jawaban soal tes uraian (*essay*) berbahasa Indonesia. Hasil pengujian memperlihatkan bahwa sistem telah dapat melakukan pengujian jawaban *essay* dengan *output* berupa sebuah nilai.
2. Penilaian jawaban yang dihasilkan sistem dipengaruhi oleh ada tidaknya term kunci pada kalimat jawaban. Semakin banyak perbedaan jumlah kata pada suatu kalimat antara guru dan siswa, maka nilai yang dihasilkan semakin kecil. Jika suatu kalimat jawaban siswa memiliki persebaran term kunci yang semakin sama dengan kunci jawaban guru maka, nilai yang dihasilkan juga semakin tinggi.

DAFTAR PUSTAKA

- [1]. Suharyanto dan A. B. L. Mailangkay, "Penerapan E-Learning Sebagai Alat Bantu Mengajar Dalam Dunia Pendidikan", *Jurnal Ilmiah Widya*, Vol. 3, No. 4, 2016.
- [2]. Ruslan, dkk, "*Implementasi Generalized Latent Semantic Analysis* Untuk Penilaian Otomatis Jawaban Esai Siswa Pada Tingkat Sekolah Menengah Atas (SMA)", *Seminar Nasional Teknologi Informasi dan Komunikasi*, 2017.

- [3]. E. S. Pramukantoro, “Sistem Penilaian Otomatis Jawaban Esai Pada Elearning Belajardisini.Com”, Jurnal Teknologi Informasi dan Ilmu Komputer, Vol. 3, No. 4, 2016.
- [4]. N. H. Ariyani, dkk, “Aplikasi Pendeteksi Kemiripan Isi Teks Dokumen Menggunakan Metode Levenshtein Distance”, Jurnal semanTIK, Vol.2, No.1, 2016.
- [5]. Y. L. Joane, dkk, “Rancang Bangun Aplikasi Deteksi Kemiripan Dokumen Teks Menggunakan Algoritma Ratcliff/Obershelp”, E-Journal Teknik Informatika Vol. 11, No. 1, 2017.
- [6]. Tinaliah dan T. Elizabeth, “Perbandingan Hasil Deteksi Plagiarisme Dokumen dengan Metode Jaro-Winkler Distance dan Metode Latent Semantic Analysis”, Jurnal Teknologi dan Sistem Komputer, 2018.
- [7]. G. Urva dan H. F. Siregar, “Pemodelan UML E-Marketing Minyak Goreng” Jurnal Teknologi Dan Sistem Informasi Vol. 1, No. 2, 2015.
- [8]. Barakbah, dkk., “Logika Dan Algoritma”, E-book Program Studi Teknik Informatika dan Komputer Politeknik Elektronika Negeri Surabaya. 2013.
- [9]. Minarni dan Susanti, “Sistem Informasi Inventory Obat Pada Rumah Sakit Umum Daerah (RSUD) Padang”, Jurnal Momentum, Vol. 16, No. 1, 2014.
- [10]. Y. B. L. Hege, dkk, “Sistem Informasi Geografis (SIG) Pelayanan Kesehatan Di Kotamadya Yogyakarta Berbasis Web”, Jurnal SCRIPT, Vol. 1, No. 2, 2014.
- [11]. A. Firman, dkk, “Sistem Informasi Perpustakaan Online Berbasis Web”, E-journal Teknik Elektro dan Komputer, Vol. 5, No. 2, 2016.