

IMPLEMENTASI ALGORITMA *JARO WINKLE DISTANCE* DALAM PROSES PENCOCOKAN JAWABAN PADA APLIKASI UJIAN

Nunu Nugraha¹, Nenden Rostika Dewi²

^{1,2}Universitas Kuningan

Jl. Cut Nyak Dhien no.36A Kuningan

Nunu.nugraha@uniku.ac.id¹, 2013081094@student.uniku.ac.id²

Abstrak

Evaluasi kegiatan belajar mengajar atau ujian dilakukan sebagai tolak ukur pemahaman siswa terhadap materi yang diberikan oleh guru. Terdapat dua jenis ujian yaitu pilihan ganda dan essay, umumnya penyelenggaraan ujian di SMP N 1 Garawangi dilakukan secara konvensional dengan menggunakan kertas dan alat tulis. Namun menurut guru mata pelajaran IPS dalam menyelenggarakan secara konvensional memerlukan waktu yang tidak sedikit dalam mengoreksi jawaban, terlebih jika soal yang diberikan adalah essay. Selain itu banyaknya jumlah siswa dan soal yang diberikan terkadang dapat menyebabkan penilaian menjadi kurang objektif dan mengakibatkan terjadinya kesalahan dalam pengoreksian jawabannya.

Berdasarkan permasalahan tersebut diperlukan adanya suatu aplikasi yang dapat mengoreksi ujian esai secara otomatis dengan menggunakan algoritma yang dapat mencocokkan jawaban berupa *string* yaitu algoritma *jaro winkler distance*. Algoritma *jaro winkler distance* memiliki kompleksitas waktu *quadratic runtime complexity* yang efektif pada *string* pendek dan dapat bekerja lebih cepat dari algoritma *edit distance*. Aplikasi ini dapat melakukan pengoreksian jawaban ujian sehingga diharapkan dengan adanya aplikasi ini dapat meminimalisir kekurangan yang ada pada ujian yang dilakukan secara konvensional.

Kata kunci: algoritma, *jaro winkler distance*, ujian, *string matching*, pencocokan jawaban

I. PENDAHULUAN

A. Latar Belakang Masalah

Setiap proses pembelajaran membutuhkan suatu evaluasi berupa ujian. Ujian dapat dilakukan dalam tiga jenis, yang pertama yaitu ujian pilihan ganda, ujian isian singkat dan ujian esai. Ujian esai merupakan evaluasi pembelajaran dalam bentuk soal esai yang mempunyai jawaban lebih bervariasi dibanding soal pilihan ganda. Variasi jawaban tersebut memberikan kesulitan terhadap guru dalam menilai jawaban esai. Ujian dengan cara ini tetap menjadi pilihan guru untuk mengevaluasi tingkat kemampuan pemahaman siswa walaupun kenyataannya tidak mudah untuk memberikan penilaian yang objektif pada jawaban siswa terlebih jika guru tersebut sudah merasa kelelahan banyaknya jawaban siswa yang harus dikoreksi.

Guru memerlukan waktu yang banyak untuk memeriksa jawaban esai, semakin banyak jumlah ujian dan banyaknya jumlah siswa yang mengikuti ujian, maka semakin banyak jumlah ujian yang harus dicocokkan / disamakan dengan kunci jawaban sehingga sering terjadi kesalahan dalam mengoreksi jawaban ujian yang menimbulkan kerugian pada sebagian siswa

karena mendapatkan nilai yang rendah serta penilaian dengan secara langsung oleh guru memerlukan waktu yang lama terlebih jika guru tersebut memiliki siswa dengan jumlah yang banyak maka akan memakan banyak waktu untuk mencocokkan jawaban dengan kunci jawaban yang tersedia sehingga penyediaan informasi nilai yang ingin siswa ketahui sedikit terhambat akibat memakan banyak waktu dalam pencocokan jawaban yang dikerjakan secara langsung oleh guru. Salah satu metode yang dapat digunakan untuk meminimalisir kekurangan yang terjadi pada koreksi soal esai otomatis yaitu dengan menerapkan algoritma *jaro winkler distance*, algoritma *jaro winkler distance* dapat mengukur kesamaan atau kemiripan dua buah *string*.

B. Rumusan masalah

Berdasarkan pada identifikasi masalah yang telah diungkapkan sebelumnya, maka permasalahan pokok yang akan dibahas dalam penelitian ini adalah:

- 1) Bagaimana merancang dan membangun suatu aplikasi ujian yang dapat memberikan penilaian yang objektif terhadap siswa?

- 2) Bagaimana cara membangun aplikasi ujian secara online?
- 3) Bagaimana cara menerapkan algoritma *jaro winkler distance* untuk pencocokan jawaban soal?

C. Batasan Masalah

Berdasarkan latar belakang dan identifikasi yang telah di ketahui, penulis membatasi permasalahan yang akan di bahas sebagai berikut :

1. Penelitian dilakukan di SMP N 1 Garawangi dalam pengkoreksian jawaban pada mata pelajaran Ilmu Pengetahuan Sosial.
2. Proses pencocokan jawaban ujian dengan mengimplementasikan algoritma *jaro winkler distance* pada pencocokan kunci jawabannya.
3. Pembuatan aplikasi pencocokan jawaban soal ujian dengan menggunakan bahasa pemrograman *web PHP* dan *database Mysql*.
4. Ujian yang dimaksud disini adalah ujian berupa ulangan harian.
5. Batas soal yang diberikan dalam sekali ujian yaitu 35 soal yang terdiri dari soal esay pendek dan soal esay uraian.
6. Aplikasi ini hanya dapat mengkoreksi jawaban berupa *string*, jika jawaban lebih dari satu maka disarankan menggunakan tanda koma (,).

D. Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penulisan penelitian ini adalah sebagai berikut:

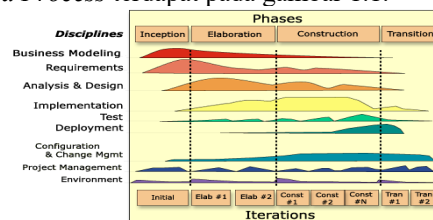
- 1) Metode Pengumpulan Data
Metode prngumpulan data yang dilakukan yaitu dengan cara ;
 - a) Observasi
Teknik pengumpulan data dengan melakukan penelitian terhadap kegiatan yang dilakukan saat kegiatan ujian, pengkoreksian jawaban dan penilaian yang berhubungan dengan pembuatan Aplikasi Pencocokan Jawaban Soal Ujian.
 - b) Wawancara
Teknik pengumpulan data dengan melakukan Tanya jawab dengan pihak bersangkutan di SMP N 1 Garawangi dalam mata pelajaran Ilmu Pengetahuan Sosial yang berhubungan dengan pembuatan Aplikasi Pencocokan Jawaban Soal Ujian.
 - c) Studi Pustaka
Teknik pengumpulan data dengan melakukan pengumpulan literatur, jurnal dan referensi-referensi yang membahas

tentang Aplikasi Pencocokan Jawaban Ujian.

2) Metode Pengembangan Sistem

Dalam pembuatan penelitian ini metode pengembangan perangkat lunak yang digunakan adalah *Rational Unified Process* (RUP) menggunakan konsep *Object Oriented*, penulis memilih metode *Rational Unified Process* (RUP) dikarenakan RUP menyediakan akses yang mudah terhadap pengetahuan dasar bagi anggota tim, menyediakan petunjuk bagaimana menggunakan UML secara efektif, mendukung proses dalam pengembangan *software*, memungkinkan adanya pe-nambahan-penambahan pada proses. Dengan adanya aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Modelling Language* (UML).[2]

Fase-fase yang terdapat dalam *Rational Unified Process* terdapat pada gambar 1.1.



Gambar 1.1 *Rational Unified Process*

Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak, yaitu:

1. Inception (Pengumpulan data)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*bussines modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*), dengan cara melakukan observasi, wawancara dan studi pustaka.

2. Elaboration (Analisis dan Desain)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang focus pada purwarupa sistem.

3. Contruction (Coding dan Implementasi)

Tahap ini lebih focus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implemantasi dan pengujian sistem yang focus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat batas/tonggak kemampuan operasional awal.

4. *Transition* (Testing dan Maintenance)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari batas/tonggak kemampuan operasional awal. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

II. LANDASAN TEORI

A. Ujian

Ujian merupakan salah satu cara untuk mengevaluasi proses belajar mengajar. Dalam dunia pendidikan, ujian dimaksudkan untuk mengukur taraf pencapaian suatu tujuan pengajaran oleh siswa atau mahasiswa sebagai peserta didik, sehingga siswa ataupun mahasiswa dapat mengetahui tingkat kemampuannya dalam memahami bidang studi yang sedang ditempu. Bila ternyata hasilnya belum maksimal, maka proses belajar harus lebih ditingkatkan baik kualitas maupun kuantitas (Walter, 2006).

Ujian konvensional atau ujian manual sudah diterapkan sejak puluhan tahun yang lalu, ujian jenis ini menggunakan alat tulis sebagai media ujian yaitu berupa kertas, pensil, pena, dan alat tulis umum lainnya untuk melaksanakan ujian. Soal ujian dan jawaban harus dijawab semuanya dan dilakukan dengan tulisan tangan.

B. Pencocokan Jawaban (*String Matching*)

String matching dalam bahasa Indonesia dikenal dengan istilah pencocokan teks, yaitu suatu metode yang digunakan untuk menemukan suatu keakuratan atau hasil dari satu atau beberapa pola teks yang diberikan. Pencocokan *string* merupakan pokok bahasan yang penting dalam ilmu komputer karena teks adalah bentuk utama dari pertukaran informasi antar manusia, misalnya pada literatur, karya ilmiah, halaman web dan sebagainya [1]. Persoalan pencarian *string* dirumuskan sebagai berikut:

- 1) Sebuah teks (*text*), yaitu sebuah (*long*) *string* yang panjangnya n karakter.
- 2) *Pattern*, yaitu sebuah *string* dengan panjang m karakter ($m < n$) yang akan dicari dalam teks. Carilah lokasi pertama di dalam teks yang bersesuaian dengan *pattern*. Aplikasi permasalahan pencocokan *string* biasa ditemukan dalam pencarian sebuah kata dalam dokumen (misalnya menu *Find* dalam Microsoft Word). Contoh :

Pattern : not

Teks : nobody noticed him

↑
Target

Dalam algoritma pencocokan *string*, teks diasumsikan berada di dalam memori, sehingga bila kita mencari *string* di dalam sebuah arsip, maka semua isi arsip perlu dibaca terlebih dahulu kemudian disimpan di dalam memori. Jika *pattern* muncul lebih dari sekali di dalam teks, maka pencarian hanya akan memberikan keluaran berupa lokasi *pattern* ditemukan pertama kali.

C. Algoritma Jaro Winkler Distance

Algoritma *Jaro Winkler Distance* merupakan varian dari *Jaro Distance* metrik yaitu sebuah algoritma untuk mengukur kesamaan antara dua string, biasanya algoritma ini digunakan didalam pendeteksian plagiat. Semakin tinggi *Jaro Winkler Distance* untuk dua *string*, maka semakin mirip dengan *string*, maka semakin mirip dengan *string* tersebut. Nilai normalnya yaitu 0 untuk menandakan tidak ada kesamaan, dan 1 untuk menandakan adanya kesamaan.[1]

Dasar dari algoritma ini sendiri memiliki tiga bagian :

1. Menghitung panjang *string*
2. Menentukan jumlah karakter yang sama didalam dua *string*

Menentukan jumlah transposisi.

Pada algoritma *Jaro Winkler Distance* digunakan rumus untuk menghitung jarak (d_j) antara dua *string* S_1 dan S_2 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
- $|S_1|$ = Panjang string 1
- $|S_2|$ = 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 menyatakan panjang prefix atau panjang karakter yang sama sampai ditemukan

ketidaksamaan (maksimum 4 karakter), maka untuk menghitung Jaro Winkler Distance (dw) adalah :

$$dw = dj + (l \times P(1 - dj))$$

Dimana :

Dj = Jaro distance untuk string S1 dan S2

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

p = Konstanta scaling factor (Nilai standar untuk konstanta ini menurut winkler adalah p = 0,1)

III. ANALISIS DAN PERANCANGAN

A. Analisis Sistem

Konsep kegiatan analisis sistem ini akan membahas mengenai algoritma yang digunakan pada sistem ini yaitu dengan menggunakan algoritma *Jaro Winkler Distance*.

Analisis sistem bertujuan untuk melakukan identifikasi persoalan-persoalan yang muncul dalam pembuatan sistem. Hal ini dilakukan agar saat analisis hasil uji coba aplikasi tidak terjadi kesalahan-kesalahan, sehingga sistem dapat berjalan dengan baik dan selesai tepat pada waktu yang telah ditentukan.

B. Analisis Kebutuhan Fungsional

Adapun kebutuhan fungsional dalam proses pengkoreksian jawaban dengan menggunakan algoritma *jaro winkler distance* diantaranya:

1. Pada sisi Guru dapat mengelola data kamus, data ujian, serta data uraian.
2. Pada sisi Siswa dapat melaksanakan ujian dan memberikan informasi hasil ujian pada saat itu juga.
3. Aplikasi ini mudah digunakan baik dari sisi guru maupun dari sisi siswa karena tampilannya yang menarik dan mudah digunakan (*userfriendly*).

B. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah tipe kebutuhan yang berisi properti perilaku yang dimiliki oleh aplikasi. Tujuan dari fase analisis ini adalah memahami dengan sebenar-benarnya kebutuhan dari aplikasi yang akan dibangun pada aplikasi ujian.

Tabel 3.1 Tabel Spesifikasi untuk PC atau Laptop (*portable computer*)

Processor	Intel® Core 2 Duo E4600 @ 2.40 GHz
Harddisk	500 GB
RAM	DDR2 1GB
VGA(GPU)	Intel HD Graphics

Monitor	12"
---------	-----

Tabel 3.2 Tabel Spesifikasi Perangkat Lunak Pembangunan Aplikasi

OS Windows 7 Ultimate
XAMPP 1.7.3
Adobe Dreamweaver CS6
Rational Rose
Microsoft Office Visio 2007
Edraw Max 7
Web Browser (Mozilla Firefox, Google Chrome)

C. Analisis Algoritma

Algoritma *Jaro Winkler Distance* merupakan varian dari *Jaro Distance* metrik yaitu sebuah algoritma untuk mengukur kesamaan antara dua string, biasanya algoritma ini digunakan didalam pendeteksian plagiat. Semakin tinggi *Jaro Winkler Distance* untuk dua string, maka semakin mirip dengan string, maka semakin mirip dengan string tersebut. Nilai normalnya yaitu 0 untuk menandakan tidak ada kesamaan, dan 1 untuk menandakan adanya kesamaan.[1] Dasar dari algoritma ini sendiri memiliki tiga bagian :

1. Menghitung panjang string
2. Menentukan jumlah karakter yang sama didalam dua string
3. Menentukan jumlah transposisi.

Berikut adalah contoh perhitungan algoritma *jaro winkler distance* :

Soal : Siapakah patih kerajaan majapahit ?

Kunci jawaban : gajah mada (S₁)

Jawaban : gadjahmada (S₂)

	g	a	d	j	a	h	m	a	d	a
g	1	0	0	0	0	0	0	0	0	0
a	0	1	0	0	0	0	0	0	0	0
j	0	0	1	0	0	0	0	0	0	0
a	0	0	0	0	0	0	0	0	0	0
h	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
m	0	0	0	0	0	0	1	0	0	0
a	0	0	0	0	0	0	0	1	0	0
d	0	0	0	0	0	0	0	0	1	0

a	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---

Dapat diketahui:

$$m = 7$$

$$S_1 = 10$$

$$S_2 = 10$$

Susunan *string* 1 (S_1) dan *string* 2 (S_2) bila diperhatikan tidak terdapat karakter yang tertukar, maka nilai transposisi (t) = 0.

Maka nilai *jaro distance* antara *string* 1 (S_1) dan *string* 2 (S_2):

$$dj = \frac{1}{3} \times \left(\frac{7}{10} + \frac{7}{10} + \frac{7-0}{7} \right) = 0.838$$

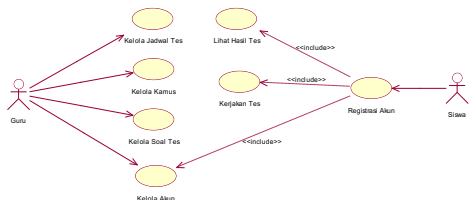
Kemudian bila diperhatikan susunan S_1 dan S_2 dapat diketahui nilai $l = 3$, dan dengan nilai konstanta $p = 0,1$ maka nilai *jaro winkler distance* adalah:

$$dw = 0,838 + (3 \times 0,1(1 - 0,838)) = 0,8866$$

D. Perancangan Sistem

1) Use Case Diagram

Use case diagram proses pengkoreksian jawaban dengan menggunakan algoritma *jaro winkler distance* dapat dilihat pada gambar



Gambar 3.1

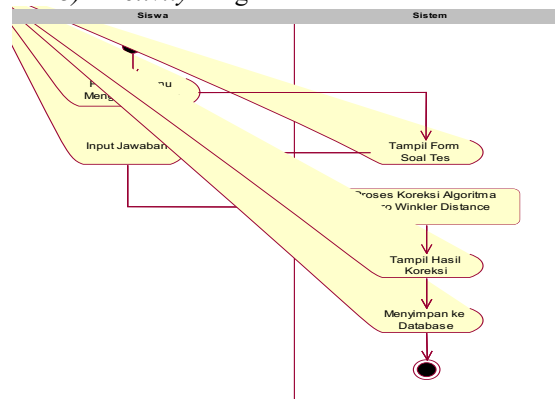
2) Skenario diagram

Tabel 3.3 Skenario Mengerjakan tes

Identifikasi	
Nama use case	Mengerjakan Tes
Tujuan	Untuk siswa melihat dan menjawab tes
Deskripsi	
Prioritas	Primary
Aktor	Siswa
Skenario Utama	
Pre Condition	Sudah terdaftar sebagai aktor dan memiliki hak akses terhadap sistem.

Aksi Aktor	Reaksi Sistem
Aktor memilih sub menu kerjakan tes	Sistem mengambil data jadwal tes dari tabel "tes_uraian" pada <i>database</i> dan menampilkan-nya.
Aktor memilih kerjakan ujian	Sistem mengambil data soal dari tabel "soal" pada <i>database</i> dan menampilkan soal tes
Aktor mengisi jawaban pada form jawaban dan menekan tombol simpan	Sistem menyimpan data jawaban kedalam tabel "jawaban" dan melakukan proses koreksi jawaban dengan menggunakan algoritma <i>jaro winkler distance</i> kemudian menghasilkan nilai hasil koreksi yang selanjutnya disimpan kedalam tabel "nilai_v_per_siswa" pada <i>database</i> dan ditampilkan kepada siswa.
Post Condition	data jawaban akan disimpan kedalam <i>database</i> pada tabel "jawaban". Dan data nilai disimpan kedalam tabel "nilai" kemudian sistem menampilkan nilai tersebut sebagai hasil dari ujian yang dilakukan siswa.

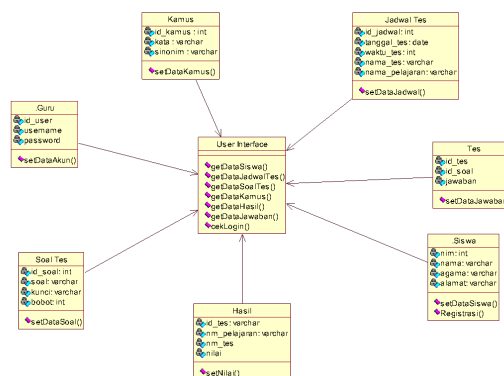
3) Activity Diagram



Gambar 3.2 Activity Diagram Mengerjakan tes

4) Class Diagram

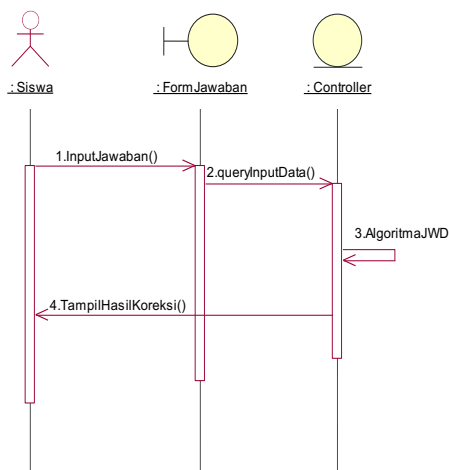
Class Diagram Aplikasi Ujian



Gambar 3.3 Class Diagram Aplikasi Ujian

5) Sequence Diagram

Sequence Diagram Mengerjakan Tes



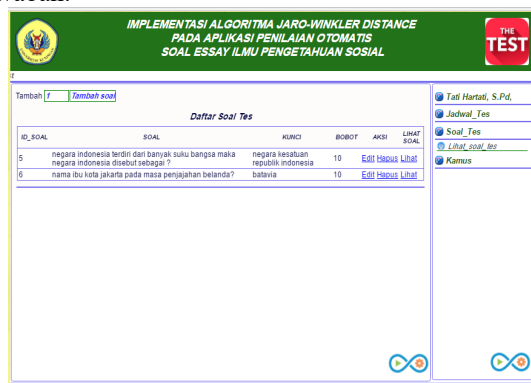
Gambar 3.4 Sequence Diagram Mengerjakan Tes
IV. IMPLEMENTASI DAN PEMBAHASAN

A. Implementasi Sistem

Implementasi perancangan pada sistem aplikasi ujian dengan dibangun dengan menggunakan bahasa pemrograman web PHP dan program dijalankan pada *web browser*. Hasil dari implementasi program pada sisi guru yaitu pengelolaan data berupa data kamus, data ujian, dan data uraian sedangkan hasil implementasi program pada sisi siswa yaitu melakukan ujian dan pada proses koreksi jawabannya menggunakan algoritma *jaro winkler distance* serta siswa dapat melihat nilai hasil ujiannya.

B. Implementasi Antarmuka

Pada implementasi antar muka terdapat dua user yaitu guru dan siswa. Pada sisi menu guru terdapat menu kelola jadwal tes, soal tes dan kelola kamus. Berikut adalah tampilan dari sub menu soal_tes pada sisi guru yang berfungsi untuk mengelola data soal dan kunci jawaban:



Gambar 4.1 Antarmuka sub menu lihat_soal_tes

Sedangkan pada sisi menu siswa terdapat menu tes_uraian yang didalamnya terdapat sub menu mengerjakan tes dan didalam sub menu tersebut pada aksi kelola terdapat fungsi dari algoritma *jaro winkler distance* yang berfungsi melakukan peng-koreksian terhadap kunci jawaban guru dan jawaban siswa. berikut adalah tampilan *output* dari sub menu mengerjakan tes pada aplikasi ujian dari sisi siswa:



Gambar 4.2 Antarmuka sub menu mengerjakan tes
Pada gambar diatas terdapat nilai hasil koreksi dan kunci jawaban dimana jawaban yang sama mendapatkan nilai 10 dan jawaban yang kurang tepat namun memiliki persamaan mendapatkan nilai 9, dengan perhitungan yang telah ditentukan menggunakan algoritma *jaro winkler distance* yang dikalikan dengan bobot yang telah ditentukan sebelumnya oleh guru.

C. Pengujian Sistem

Pengujian sistem adalah proses mengeksekusi perangkat lunak untuk menentukan apakah sistem tersebut memenuhi spesifikasi sistem dan berjalan sesuai dengan yang diharapkan

a) Pengujian *black box*

Pengujian *black box* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, penguji atau *tester* akan mendefinisikan dan melakukan pengujian terhadap spesifikasi fungsional program.

Tabel 1 pengujian *black box*

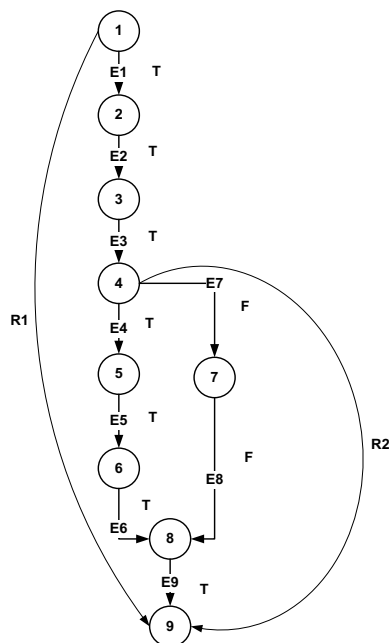
Kasus dan Hasil Uji (Benar)			
Item Uji	Yang diharapkan	Pengamatan	Kesimpulan
Memilih menu tes_uraian	Menampilkan sub menu kerjakan_tes dan lihat_hasil_tes	Berhasil menampilkan sub menu kerjakan_tes dan lihat hasil tes	Valid
Memilih sub menu kerjakan_tes	Menampilkan jadwal tes yang tersedia dan terdapat aksi kerjakan	Berhasil menampilkan jadwal tes yang tersedia dan terdapat aksi kerjakan	Valid
Memilih aksi kerjakan	Menampilkan form daftar soal tes dan terdapat satu buah <i>button</i> simpan	Berhasil menampilkan daftar soal tes dan terdapat satu buah <i>button</i> simpan	

Menekan <i>button</i> simpan	Menyimpan jawaban kedalam <i>database</i> dan soal yang sudah terisi jawaban terhapus dari daftar soal yang tersedia	Berhasil menyimpan jawaban kedalam <i>database</i> dan soal yang sudah terisi jawaban terhapus dari daftar soal yang tersedia	
Memilih sub menu lihat_hasil_tes	Masuk ke halaman sub menu lihat_hasil_tes dan menampilkan tabel daftar nilai tes yang telah dikerjakan serta terdapat satu buah aksi detail jawaban.	Berhasil masuk ke halaman sub menu lihat_hasil_tes dan menampilkan tabel daftar nilai tes yang telah dikerjakan serta terdapat satu buah aksi detail jawaban.	Valid

b) Pengujian *white box*

Pengujian White Box berfokus pada structural kontrol program. Adapun metode yang digunakan dalam pengujian white box ini adalah metode basis Path untuk memastikan bahwa semua statement pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa kondisi logis yang telah diuji.

Berikut ini adalah gambar *Flow Graph Notation* dalam melakukan *white box testing*:



Gambar 5.1 Flowgraph Algoritma Jaro Winkler Distance

- Keterangan gambar :
 Region (R) : 2
 Node (N) : 9
 Edge (E) : 9
 Predicate Node (P) : 1
 T : True

F : False

2. Perhitungan

$$a. V(G) = E - N + 2$$

Dimana :

$$E = 9$$

$$N = 9$$

$$V(G) = 9 - 9 + 2$$

$$= 2$$

$$b. V(G) = P = 1$$

Dimana :

$$P = 1$$

$$V(G) = 1 + 1 = 2$$

$$c. Cyclomatic Complexity$$

$$R1, R2 = 2$$

Jadi *Cyclomatic Complexity* untuk *flow graph* ini = 2

3. Langkah Alur

Path 1 : 1 - 2 - 3 - 4 - 5 - 6 - 8 - 9

Path 2 : 1 - 2 - 3 - 4 - 7 - 8 - 9

Setelah melakukan langkah-langkah diatas dalam melakukan *white box testing*, berikut adalah hasil *white box testing* ada pada tabel berikut

Pengujian White box

Tabel 4.9 Pengujian *White Box*

No	Cara Pengujian	Langkah Alur	Hasil yang diharapkan	Hasil yang keluar	Kesimpulan
1	Menginputkan jawaban benar	1 - 2 - 3 - 4 - 5 - 6 - 8 - 9	Sistem akan mem-proses dan menampilkan nilai yang sesuai	Sesuai harapan	Valid
3	Menginputkan jawaban salah	1 - 2 - 3 - 4 - 7 - 8 - 9	Sistem akan menghitung nilai minimum dari nilai kecocokan minimum	Sesuai harapan	Valid

V. PENUTUP

A. Kesimpulan

Berdasarkan hasil analisa dan perancangan serta pemanfaatan algoritma *jaro winkler distance* pada aplikasi ujian di SMP N 1 Garawangi, maka diperoleh kesimpulan sebagai berikut :

- Aplikasi ini dapat melakukan penyelenggaraan ujian dengan tipe soal essay dan proses pengkoreksiannya menggunakan sistem. Serta siswa dapat mengisi soal dan pada saat itu juga siswa dapat langsung memperoleh nilai dari ujian yang telah dilaksanakannya.

2. Dengan memanfaatkan algoritma *jaro winkler distance* pada aplikasi ujian dapat membantu mempermudah guru dalam melakukan proses pengkoreksian jawaban.
3. Aplikasi ini dapat memberikan informasi detail jawaban yang didalamnya terdapat soal, jawaban siswa dan kunci jawaban yang sebelumnya telah disediakan oleh guru serta nilai yang diperoleh siswa dari tes yang telah dilakukan.
4. Dengan aplikasi ini siswa dapat mengetahui dimana kesalahan jawabannya karena aplikasi ini dapat memunculkan kunci jawaban yang telah disediakan oleh guru pada detail jawaban yang ada pada tabel lihat hasil tes

B. Saran

Setelah penulis mengadakan penelitian, maka ada beberapa saran yang ingin penulis kemukakan agar aplikasi berjalan maksimal, antara lain:

1. Untuk pengembangan aplikasi ujian ini jika terdapat admin untuk mengelola data akun baik guru dan juga siswa sehingga dapat menghasilkan pengelolaan data yang lebih baik,
2. Dalam pengelolaan nilai jika terdapat modul untuk mengelola nilai dapat memudahkan guru dalam pengelolaan nilai dalam pembuatan laporan akhir semester,
3. Program ini akan lebih baik jika terdapat modul remedial bagi siswa yang memiliki nilai hasil tes kurang dari standar yang ditetapkan oleh guru,

Jika terdapat lebih dari satu mata pelajaran pada aplikasi ujian ini dapat dikembangkan menjadi *e-learnig*.

DAFTAR PUSTAKA

- [1] Anna Kurniawati, Sulisty Puspitodjati, Sazali Rahman(2010), Implementasi Algoritma Jaro-Winkler Distance Untuk Membandingkan Kesamaan Dokumen Berbahasa Indonesia. Depok :Universitas Gunadarma
- [2] A.S Rosa & M. Salahuddin. (2013), *Modul Pembelajaran Rekayasa Perangkat Lunak*, Bandung : Informatika
- [3] Kornain, Ahmad, Ferry Yansen (2010), *Penerapan Algoritma Jaro-Winkler Distance Untuk Sistem Pendeteksi Plagiarisme Pada DokumenTeks Berbahasa Indonesia*, Program Studi Teknik Informatika, STIMIK GI MDP.