Fuzzy String Matching for Semi-Automatication of Words with Jaro Winkler Distance Algorithm on Microsoft Word Documents

Fuzzy String Matching untuk Semi-Otomatisasi Pencocokan Kata dengan Algoritma *Jaro Winkler Distance* pada Dokumen *Microsoft Word*

Hasna Nur Hanani[1], Herlina Jayadianti[2], Heru Cahya Rustamaji[3]

[1,2,3] Informatika, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia

[1]hasnanani@gmail.com, [2*]Herlina.jayadianti@upnyk.ac.id, [3]herucr@gmail.com

**:Penulis korenspondensi (corresponding author)*

Keywords:matching;semi-automation;fuzzy string matching; jaro winkler distance; foreign word

Abstract

Purpose:measuring effect of fuzzy string matching on the italicizing of foreign words in semi-automatication with jaro winkler distance on number of words, time and accuracy.
Design/method/approach: testing accuracy and time of processing foreign word italicizing with Jaro Winkler and testing accuracy and time of processing foreign word italicizing with additional fuzzy string matching.
Findings/result: The semi-automation accuracy of words in the first data test resulted in a value of 83,73% for italicizing with the Jaro Winkler distance algorithm and 84.33% for word italicizing with fuzzy string matching while the semi-automated word on data test two with the Jaro Winkler algorithm was 98, 77%, while with the addition of fuzzy string matching the accuracy becomes 99.11%. Measurement of processing time shows that the addition of fuzzy string matching tends to be faster. The number of words italicized with fuzzy string matching is more than the word skewing with jaro winkler distance in conditions where the number of foreign words in the database is the same.
Originality/value/state of the art: This research begins with the detection of foreign words in documents. If a foreign word which is an English word is found, then in the first test the word will be measured by the Jaro Winkler distance algorithm. Jaro Winkler is used to measure the word similarity between foreign words found in documents and foreign words in the database. If the result of the jaro

ISSN: 1979-2328

winkler distance is 1 then the word will be italicized. In the second test, the foreign words that have been measured by Jaro Winkler will have a word equivalent value which will be processed by fuzzy string matching. Fuzzy string matching will give a tolerance value to the result of the word equation value. The result of fuzzy string matching value will determine whether the foreign word will be italicized or not.

Abstrak

Kata kunci: pencocokan; semi-otomatisasi; tiga; *fuzzy string matching; jaro winkler distance*; kata asing

Tujuan: mengukur pengaruh keberadaan *fuzzy string matching* pada pemiringan kata asing secara semi-otomatisasi dengan *jaro winkler distance* terhadap jumlah kata, waktu dan akurasi.

Perancangan/metode/pendekatan: menguji ketepatan dan waktu pemrosesan pemiringan kata asing dengan *jaro winkler* dan menguji ketepatan dan waktu pemrosesan pemiringan kata asing dengan tambahan *fuzzy string matching.*

Hasil: Akurasi semi-otomatisasi kata pada uji data satu menghasilkan nilai 83,73% untuk pemiringan dengan algoritma *jaro winkler distance* dan 84,33% untuk pemiringan kata dengan *fuzzy string matching* sedangkan semi-otomatisasi kata pada uji data dua dengan algoritma *jaro winkler* adalah 98,77%, sedangkan dengan penambahan *fuzzy string matching* akurasinya menjadi 99,11%. Pengukuran waktu pemrosesan menunjukkan bahwa dengan penambahan *fuzzy string matching* cenderung lebih cepat. Jumlah kata yang dimiringkan dengan *fuzzy string matching* lebih banyak dibanding pemiringan kata dengan *jaro winkler distance* pada kondisi dimana jumlah kata asing pada *database* sama.

Keaslian/ *state of the art*: Penelitian ini, bermula dari pendeteksian kata asing pada dokume. Apabila kata aing yang merupakan kata dalam bahasa Inggris ditemukan, maka pada pngujian pertama kata tersebut akan diukur dengan algoritma *Jaro Winkler distance*. Jaro Winkler digunakan untuk mengukur persamaan kata antara kata asing yang ditemukan pada dokumen dengan kata asing pada *database*. Jika hasil dari jaro winkler distance adalah 1 maka kata akan dicetak miring. Pada pengujian kedua kata asing yang telah diukur dengan jaro winkler akan memiliki nilai persamaan kata yang akan diolah oleh *fuzzy string matching*. Fuzzy string matching akan memberikan toleransi nilai kepada hasil nilai persamaan kata. Hasil nilai *fuzzy*

string matching akan menentuka kata asing tersebut akan dicetak miring atau tidak .

1. Pendahuluan

Gorys Keraf menyatakan bahwa kata merupakan bagian terkecil yang mengandung sebuah ide dari suatu susunan kalimat [1]. Apabila kata disusun secara sistematis akan mengandung sebuah pesan, pemikiran atau hasil/kesimpulan. Rangkaian kata yang ditulis dapat disebut sebagai karya tulis. Salah satu peraturan dalam kepenulisan adalah memiringkan kata asing pada kata yang digunakan oleh penulis. Tujuan dari pemiringan kata asing adalah untuk menghindari kerancuan pembaca dalam memahami isi teks. Kata asing yang digunakan sebagai objek penelitian ini adalah kata asing yang berbahasa inggris dan berkaitan dengan teknologi dan berasal dari www.myvocabulary.com serta kata asing berbahasa inggris pada dokumen uji coba. Kata asing tersebut akan dimasukkan ke dalam database. Pada penelitian ini, *User* juga dapat menambah, mengurangi, mengubah dan menghapus kata asing pada database. Sehingga, aplikasi yang dihasilkan bersifat semi-automasi karena masih membutuhkan bantuan *user*.

Langkah pemiringan kata asing dapat diawali dengan menerapkan metode *approximate string search* yaitu metode pencarian beberapa kata yang mengijinkan munculnya *error*[2]. Algoritma pada metode *approximate string search,* diantaranya adalah algoritma *Lavhenstein Distance*, algoritma *Damereu-Levhenstein Distance*, algoritma *Hamming Distance* dan algoritma *Jaro-Winkler Distance*. Algoritma *Jaro-Winkler Distance* dapat digunakan untuk pencocokan string dan memberikan hasil terbaik pada pencocokan dua string singkat [3]. Menurut Winkler, langkah pada algoritma pada *jaro winkler distance* cukup singkat karena teridiri dari tiga komponen dasar yaitu menghitung panjang string atau kata, mencari nomor huruf pada kedua kata, dan mencari transposisi. Semakin tinggi jarak *Jaro Winkler Distance* antara dua teks berarti semakin ada kemiripan [4].

Penelitian mengenai pendeteksian kata asing sudah pernah dilakukan dengan menerapkan *fuzzy search* dan algoritma *lavenshtein* dengan tingkat kakurasian sebessar 89%. Namun, dari hasil penelitian tersebut disebutkan bahwa aplikasi pemiringan kata asing yang telah dibangun apabila data yang dimasukkan terlalu besar maka proses pendeteksian akan membutuhkan waktu yang lama [5]. Pada penelitian tersebut user tidak dapat melakukan perubahan pada database yang berisikan kata asing. Penelitian mengenai keberadaan fuzzy string matching pada aplikasi chatbot digunakan sebagai penalaran kalimat agar lebih mudah untuk pencarian *keyword* [6]. Penerapan fuzzy string matching pada chatbot untuk menjwab pertanyaa-pertanyaan islam menghasilkan nilai 70.37% hal tersebut juga dipengaruhi oleh banyaknya kosa kata pertanyaan pada database mengenai sholat dan zakat [7].

Penelitian penerapan algoritma *jaro winkler distance* untuk pengoreksian kesalahan kata menghasilkan bahwa algoritma Jaro-Winkler dapat melakukan pengoreksian kata selama 8,5 detik pada 88 halaman pada dokumen sebuah skripsi dengan tingkat keakurasian 77,23 %. Namun, pada aplikasi tersebut belum terdapat aplikasi perbaikan kamus. Seperti fungsi tambah dan ubah kata pada kosakata yang terdapat pada kamus [8]. Dalam penelitian penerapa metode j*aro winkler distance* pada sistem *chatbot* menjelaskan bahwa dengan penambahan algoritma jaro winkler distance untuk perbaikan kata menghasilkan nilai akurasi 96% saat pemebrian respons, sedangkan jika tanpa penambahan metode jaro winkler sistem chatbot menghasilkan nilai akurasi 36%.[9]. Perbandingan nilai akurasi antara penggunaan algoritma jaro winkler dan

ISSN: 1979-2328

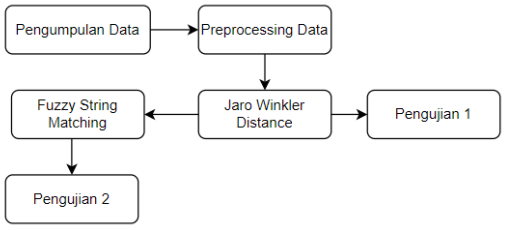
Latent Semantic Analysis untuk plagiarisme dokumen dengan data uji coba dokumen yang sama persis menghasilkan nilai akurasi 100% untuk penerapan dengan jaro winkler. Sedangkan, cek plagiarisme dokumen dengan Latent Semantic Analysis memiliki nilai akurasi 97,4%. Namun, Latent Semantic Analysis akan memberikan hasil yang baik padad dokumen yang sudah dimodifikasi [10].

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*. Teori himpunan *fuzzy* merupakan kerangka matematis yang digunakan untuk mempresentasikan ketidaktpastian, ketidakjelasan, ketidaktepatan, kekurangan informasi dan kebenaran parsial [11]. Konsep logika fuzzy mudah dimengerti karena memiliki langkah perhitungan matematis yang mendasari penalaran. Perhitungan logika fuzzy memiliki nilai toleransi dan hasil yang fleksibel. Keberadaan logika fuzzy ditujukan untuk memberikan toleransi pada saat pendeteksian kata agar kata yang dimiringkan lebih banyak daripada tanpa adanya penambahan *fuzzy string matching*. *Fuzzy string matching* merupakan pencocokan string secara samar, dimana string yang sedang dicocokkan memiliki kemiripan penulisan atau kemiripan pengucapan [12]. Nilai variable fuzzy akan dihitung dengan metode fuzzy mamdani karena fuzzy mamdani memiliki nilai mean square error dan mean absolute presentation error yang kecil [13].

Penelitian ini berfokus pada pengaruh keberadaan *fuzzy string matching* pada pemiringan kata asing dengan algoritma *jaro winkler distance.* Pengukuran kemampuan akan diukur pada tiga komponen yaitu kemampuan dari waktu yang dibutuhkan selama pemrosesan, banyaknya kata yang mampu dimiringkan dan rata-rata tingkat akurasi ketepatan pemiringan kata pada dokumen uji coba.

2. Metode/Perancangan

Metode penelitian yang digunakan adalah metode penelitian kuantitatif. Tahapan yang dilakukan dari penelitian yaitu pengumpulan data, *preprocessing* data, penerapan *jaro winkler distance*, pengujian 1, *fuzzy string matching* dan Pengujian 2.



Gambar 1. Tahapan Penelitian

2.1. Pengumpulan Data

Langkah yang digunakan untuk mendukung penelitian ini adalah pengumpulan data kata asing, 10 dokumen uji coba, dan hasil *survey* yang telah dilakukan dengan hasil 104 data responden. Dari 104 data responden tersebut menghasilkan kesimpulan bahwa keseluruhan responden setuju dengan keberadaan aplikasi pemiringan kata asing secara otomatis dengan beberapa alasan dapan membantu tugas, mempermudah pekerjaan dan menghemat waktu.

ISSN: 1979-2328

Gambar 2. Respon Responden Terhadap Keberadaan Semi-Otomatisasi Kata Asing

Data dokumen uji coba berasal dari data milik responden yang mengisi *survey*. Data dokumen uji coba yang digunakan adalah :

Tabel 1. Data Dokumen Uji Coba

No.	Nama Dokumen	Jumlah Kata
1.	Dokumen 1	2391
2.	Dokumen 2	6466
3.	Dokumen 3	3536
4.	Dokumen 4	3067
5.	Dokumen 5	4140
6.	Dokumen 6	912
7.	Dokumen 7	2452
8.	Dokumen 8	3212
9.	Dokumen 9	1130
10.	Dokumen 10	571

Dokumen ke 1 hingga 8 merupakan dokumen yang terususun dari kata asing dan kata dalam bahasa Indonesia, dokumen ke 9 merupakan dokumen yang seluruh katanya tersusun dari bahasa Indonesia, dan dokumen ke 10 merupakan dokumen yang seluruhnya tersusun oleh kata asing.

2.2. Perpocessing Data

Pada tahap ini *preprocessing* data akan membagi memisahkan kata pada setiap dokumen dengan spasi dan menghilangkan setiap tanda baca pada kata.

2.3. Jaro Winkler Distance

Algoritma *Jaro Winkler* adalah algoritma untuk mengukur kesamaan antara dua *string*, biasanya algortima ini digunakan untuk mendeteksi duplikat. Dasar dari algoritma *Jaro Winkler* memiliki tiga bagian :

1. Menghitung panjang *string*
2. Menemukan jumlah karakter yang sama di dalam dua *string*
3. Menemukan jumlah transposisi [14]

Rumus untuk menghitung jarak (dj) antara dua string (S1 dan S2) pada algoritma *Jaro Winkler Distance :*

$$d_j = \frac{1}{3} \times \left(\frac{m}{s_1} + \frac{m}{s_2} + \frac{m-t}{m} \right) \quad (1)$$

m = jumlah karakter yang sama
$|s_1|$ = Panjang string 1. String 1 merupakan string yang terdapat di dalam dokumen.
$|s_1|$ = Panjang string 2. String 2 adalah kata asing yang berada di dalam *database*
t = jumlah transposisi

Jarak teoritis dua buah karakter yang dianggap sama dikatakan benar jika tidak melebihi batas dari persamaan :

$$\text{jarak teoritis} = \left(\frac{\max(|s_1|, |s_2|)}{2} \right) < -1 \quad (2)$$

Jaro winkler (dw) menggunakan skala prefix (p) yang memberikan tingkat penilaian yang lebih, prefix panjang(l) yang menyatakan panjang awalan yaitu panjang karakter yang sama dari string yang dibandingkan sampai ditemukan ketidaksamaan. Rumus *Jaro Winkler distance* yaitu:

$$d_w = d_j + (lp(1 - d_j)) \quad (3)$$

d_j = *Jaro distance* untuk strings s1 dan s2
l = panjang *prefix* umum di awal string nilai maksimalnya 4 karakter (panjang karakter yang sama sebelum ditemukan ketidaksamaan maksimal 4).
p = konstanta *scaling* faktor. Nilai standar untuk konstanta ini menurut *Winkler* adalah p =0,1.
Threshold merupakan tahap dimana sesuatu mulai terjadi atau berpengaruh atau dapat juga diartikan sebagai titik sebelum situasi baru, preiode kehidupan atau lainnya dimulai [15]. Nilai *threshold* berupa angka 0 hingga 1. Dalam penelitian ini, nilai *threshold* yang digunakan adalah 0.91.

2.4. Fuzzy String Matching

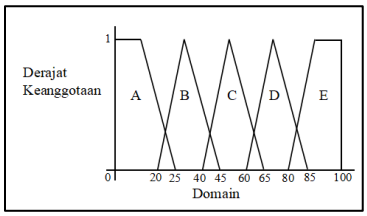
Fuzzy string matching juga dikenal dengan istilah algoritma pencarian *string. Fuzzy string matching* merupakan algoritma yang memiliki fungsi *similarity (similarity function)* yang digunakan untuk memutuskan hasil *string* pencarian dengan *string* hasil pedekatan yang terdapat di *database*[4]. *Fuzzy string matching* dalam penelitian ini, memiliki lima variabel yang dapat digunakan untuk penamaan grup dari kondisi tertentu. Nilai pada variabel *fuzzy string matching* merupakan besaran nilai dari algoritma *jaro winkler distance* hasil dari pengukuran pendeteksian antara dua kata. Yaitu kata pada dokumen dengan kata pada *database.* Nilai *jaro-winkler distance* dilambnagkan dengan variabel x. Lima variabel yang digunakan adalah tidak mirip, kurang mirip, cukup mirip, mirip dan sangat mirip[4]. Nilai dari setiap variabel dapat dilihat pada Tabel 2. Batasan penggunaan *fuzzy String matching* hanya sampai pada variabel *fuzzy*.

Tabel 2. Nilai Variabel *Fuzzy*

No	Variabel	Nilai
1	Tidak mirip	0-25
2	Kurang mirip	20-45
3	Cukup mirip	40-65
4	Mirip	60-85
5	Sangat mirip	80-100

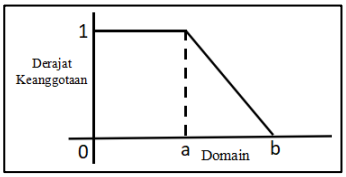
 ISSN: 1979-2328

Perhitungan setiap variabel *fuzzy string matching* menggunakan perhitungan seperti halnya yang terdapat pada *fuzzy logic* mamdani. Penggunaan langkah *fuzzy* hanya akan digunakan sampai pada tahap perhitungan himpunan *fuzzy* pada setiap variabel *fuzzy string matching*. Gambaran kurva pembagian variable *fuzzy* dapat dilihat pada gambar 3. Gambar 3 merangkum pembagian lima nilai variabel *fuzzy*.



Gambar 3. Kurva Variabel *Fuzzy String Matching*

Gambar 3 merupakan kurva yang terdiri dari 3 macam kurva yaitu kurva bahu kiri, kurva segitiga dan kurva bahu kanan.

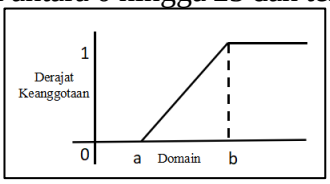


Gambar 4. Kurva Bahu Kiri

Fungsi keanggotaan variabel *Fuzzy String Match* bahu kiri :

$$\mu[x] = \begin{cases} 1; & x \leqslant a \\ (b-x)/(b-a); & a \leqslant x \leqslant b \\ 0 & x \geqslant b \end{cases} \quad (4)$$

Kurva bahu kiri berlaku untuk nilai antara 0 hingga 25 dan termasuk dalam variabel tidak mirip.

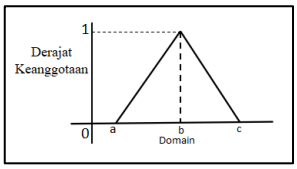


Gambar 5. Kurva Bahu Kanan

Fungsi keanggotaan variabel *Fuzzy String Matching* bahu kanan :

$$\mu[x] = \begin{cases} 0; & x \leqslant a \\ (x-a)/(b-a); & a \leqslant x \leqslant b \\ 1 & x \geqslant b \end{cases} \quad (5)$$

Kurva bahu kanan berlaku untuk nilai antara 85 hingga 100 dan termasuk dalam variabel sangat mirip.



Gambar 6. Kurva Segitiga

Fungsi keanggotaan variabel *Fuzzy String Matching* segitiga untuk variabel kurang mirip, cukup mirip, dan mirip :

$$\mu[x] = \begin{cases} 0; & x \leqslant a \text{ atau } x \geqslant c \\ (x-a)/(b-a); & a \leqslant x \leqslant b \\ (b-x)/(c-b;) & b \leqslant x \leqslant c \end{cases} \quad (6)$$

Sedangkan jikai nilai x adalah nilai antara 90 hingga 100 maka hasil perhitungan *fuzzy string Match* adalah 1 dengan kategori kurva bahu kanan. Semi otomatisasi dengan *fuzzy string matching* akan menggunakan hasil perhitungan *fuzzy* untuk menentukan sebuah kata termasuk dalam kata asing atau bukan. Nilai dari perhitungan kata jaro winkler akan dikalikan dengan 100 agar tidak membentuk bilangan desimal. Kemudian, nilai tersebut akan di masukkan ke dalam variabel *fuzzy*.

Tabel 3. Penentuan Kemiringan Kata

No	**String 1**	**String 2**	**Nilai *Jaro Winkler Distance***	**Keputus-an**	**Nilai *Jaro Winkler Distance* * 100**	**kategori**	**Nilai *Fuzzy String Matching***	**Kep-utus a-n**
1	developer	developer	1.0	*Italic*	100	Sangat Mirip	1	*italic*
2	features	feature	0.98	Normal	98	Sangat Mirip	1	*italic*

Pada pemringan kata asing dengan *fuzzy string matching* kata *developer* dan kata *features* dianggap kata asing karena nilai *fuzzy string matching* adalah 1. Sehingga, kata *features* dan kata *developer* akan dicetak miring.

2.5. Confusion Matrix

Pengujian 1 dan pengujian 2 merupakan pegujian akurasi ketepatan pada kata asing yang dimiringkan menggunakan perhitungan *confusion matrix*

Tabel 4. Perhitungan *Confusion Matrix*

Confusion Matrix		Actual Values	
		1 (Positive)	**0 (Negative)**
Predicted Values	1 (*Positive*)	TP	FP
	0 (*Negative*)	FN	TN

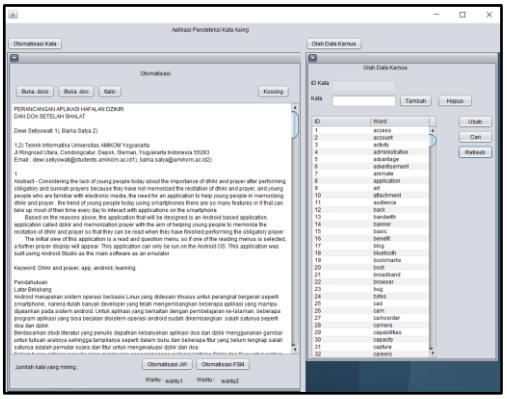
$$Akurasi = \frac{(TP + TN)}{(TP + TN + FN + FP)} \times 100\% \quad (7)$$

True Positive(TP) merupakan kondisi apabila kata asing yang terdapat dalam dokumen dan terdapat juga di kamus dicetak miring. *False Positive*(FP) terjadi apabila ada kata yang tidak terdapdat pada *dat-abase* dicetak miring. *False Negative*(FN) apabila ada kata asing yang sesuai di kamus dan di dokumen tidak dicetak miring setelah diproses oleh aplikasi. *True Negative*(TN) adalah kondisi apabila ada kata yang tidak terdapat pada *database* atau bukan kata asing dan tidak dicetak miring. Untuk mengetahui rata-rata akurasi menggunakan rumus :

$$Rata - rata = \frac{\sum akurasi\, setiap\, dokumen}{\sum banyak\, dokumen} \quad (8)$$

3. Hasil dan Pembahasan

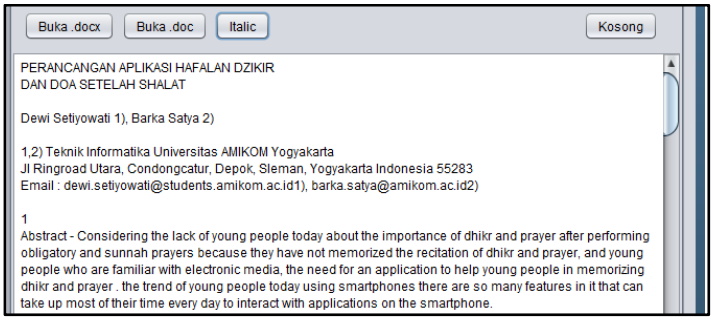
Bagian hasil dan pembahasan memuat pencatatan waktu selama pemrosesan semi-otomatisasi pendeteksian kata asing, jumlah kata asing pada dokumen yang mampu dimiringkan oleh aplikasi dan nilai akurasi setiap dokumen. Pengujian dilakukan sebanyak dua kali pada uji data satu dan uji data dua. Pada uji data satu, *database* yang digunakan adalah *database* yang berisikan kata asing dari *www.myvocabulary.com* dengan tema teknologi. Sedangkan untuk uji data dua, *database* berisikan kata asing pada *database* uji data satu ditambah kata asing yang terdapat pada dokumen uji coba yang dimasukkan secara manual ke *database*. Proses uji data 1 sama dengan proses uji data 2. Namun, kata asing pada uji data 2 lebih banyak dari pada uji data 1. Pengujian dilakukan secara satu per satu setiap dokumen. Setiap kata di dalam dokumen juga dicocokkan secara satu persatu. Dibawah ini merupakan penjelasan hasil dari Dokumen 1 saat uji data 2.



Gambar 7. Tampilan Aplikasi

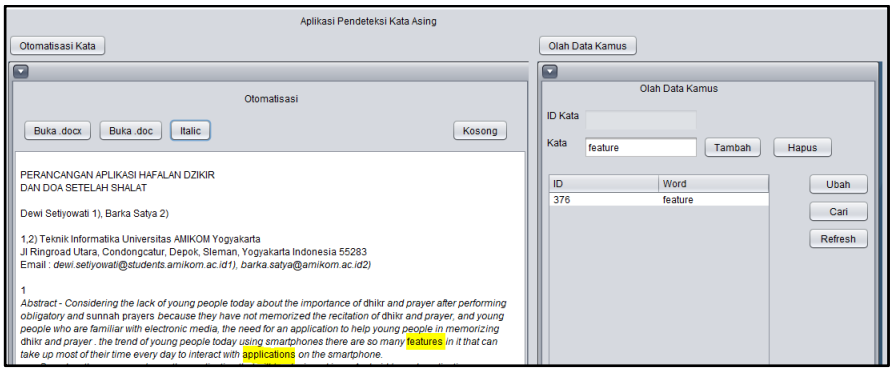
ISSN: 1979-2328

Pada gambar 7 terdapat dua bagian. Bagian sebelah kiri merupakan bagian untuk halaman otomatisasi kata dimana pada halaman tersbut isi dokumen 1 ditampilkan. Aplikasi ini mampu membuka dokumen dalam bentuk eksistensi .doc atau .docx. Bagain bawah halaman otomatisasi terdapat tombol otomatisasi dengan *jaro winkler* dan tombol otomatisasi dengan *fuzzy string matching*, serta pencatatan waktu dan jumlah kata yang mampu dimiringkan. Bagian kanan merupakan halaman olah data kata, dimana *user* dapat menambah, mengubah dan menghapus kata. Untuk memperjelas hasil proses dibawah ini terdapat bagian *abstract* dari dokumen 1.



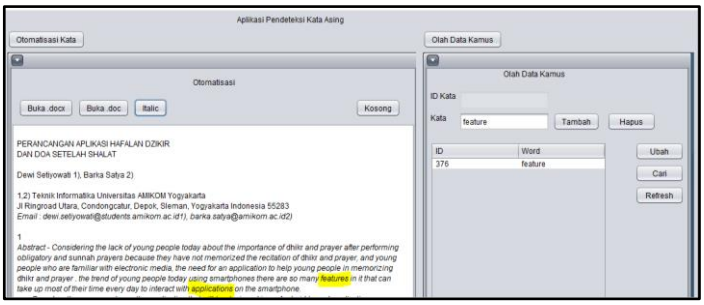
Gambar 8. Potongan Tampilan Dokumen 1 Sebelum di proses

Adapun langkah yang digunakan untuk menampilkan halaman dokumen adalah dengan memilih dokumen yang akan ditampilkan setelah menekan tombol buka .docx atau buka .doc sesuai dengan eksistensi dokumen yang ingin dibuka. Kemudian, aplikasi akan menampilkan isi dokumen dengan bantuan *library Apache POI.*



Gambar 9. Kondisi Sebelum Pendeteksian dengan *Jaro Winkler Distance*

Pada gambar 9 terdapat kata asing yang mampu dimiringkan oleh *jaro winkler distance* dan terdapat dua kata yang tidak mampu dimiringkan oleh jaro winkler. Penjelasan kata *features* dan *applications* tidak dimiringkan karena nilai kedekatan kata dengan kata pada *database* bukan 1. Berbada halnya pada pemiringan kata dengan *fuzzy string matching.*



Gambar 10. Hasil Pendeteksian dengan *Jaro Winkler Distance*

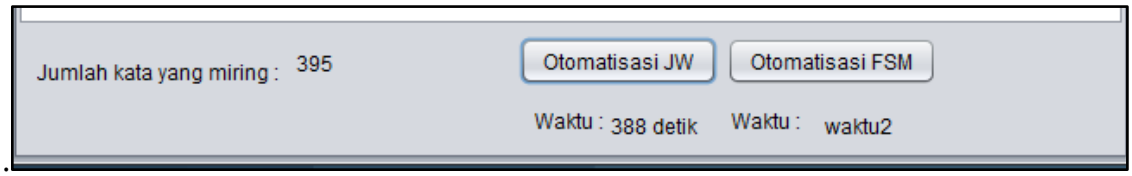
Pada gambar 10, pemiringan dengan *fuzzy string matching* mampu memiringkan kata asing yang belum bisa dimiringkan oleh *jaro winkler* sebelumnya sekalipun dengan data kata pada *database* yang sama. Hal ini menunjukan bahwa pemiringan kata asing dengan *fuzzy string matching* mampu mencetak miring kata asing lebih banyak dari pada pemiringan kata asing dengan *jaro winkler distance*.

Tabel 5. Hasil Pencatatan Waktu Uji Data 1 dan Uji Data 2

No.	Dokumen	Uji Data 1		Uji Data 2	
		Waktu pemiringan dengan *Jaro winkler Distance* (detik)	**Waktu pemiringan dengan *Fuzzy String Matching* (detik)**	**Waktu pemiringan dengan *Jaro winkler Distance* (detik)**	**Waktu pemiringan dengan *Fuzzy String Matching* (detik)**
1.	Dokumen 1	79	78	388	264
2.	Dokumen 2	213	209	604	573
3.	Dokumen 3	102	100	243	239
4.	Dokumen 4	106	101	251	234
5.	Dokumen 5	132	130	189	145
6.	Dokumen 6	32	32	82	82
7.	Dokumen 7	81	80	190	186
8.	Dokumen 8	114	113	268	264
9.	Dokumen 9	43	43	101	95
10.	Dokumen 10	49	40	33	30

Pada dokumen satu, uji data 1 membutuhkan waktu pemiringan kata asing dengan *jaro winkler distance* selama 79 detik dan 78 detik untuk memiringkan kata asing dengan *fuzzy string matching.* Sedangkan, pada uji data 2, dokumen satu membutuhkan waktu pemiringan kata asing dengan *jaro winkler distance* selama 388 detik dan 264 detik untuk pemiringan kata asing dengan *fuzzy string matching.* Begitu seterusnya hingga dokumen 10. Hasil pencatan waktu pada uji data 2 mengahsilkan nilai waktu lebih lama dibandingkan pada uji data satu. Hal tersebut dikarenakan, data kata asing pada *database* uji data 2 lebih banyak dibandingkan data kata asing pada uji data 1. Sehingga kegiatan pengukuran kecocokan kata pada uji data 2 menjadi lebih banyak. Rata-rata pencatatan waktu pada semi otomatisasi kata asing dengan *fuzzy string matching* sedikit lebih cepat dibanding semi otomatisasi kata asing dengan *jaro winkler distance*

 ISSN: 1979-2328



Gambar 11. Hasil Nilai Pemrosesan dengan *Jaro Winkler DIstance*

Gambar 11 merupakan bagian bawah dari halaman otomatisasi yang menampilkan hasil pemrosesan saat pendeeksian kata asing dengan *Jaro Winkler Distance* pada dokumen 1.

Gambar 12. Hasil Nilai Pemrosesan dengan *Fuzzy String Matching*

Gambar 12 merupakan bagian bawah dari halaman otomatisasi yang menampilkan hasil pemrosesan saat pendeeksian kata asing dengan *Fuzzy String Matching* pada dokumen 1.

Tabel 6. Hasil Perbedaan Jumlah Kata yang Dimiringkan

No.	Dokumen	Uji Data 1		Uji Data 2	
		Jumlah kata yang dimiringkan dengan *Jaro winkler Distance*	**Jumlah kata yang dimiringkan dengan *Fuzzy String Matching***	**Jumlah kata yang dimiringkan dengan *Jaro winkler Distance***	**Jumlah kata yang dimiringkan dengan *Fuzzy String Matching***
1.	Dokumen 1	74	85	395	423
2.	Dokumen 2	146	174	1045	1146
3.	Dokumen 3	76	85	259	288
4.	Dokumen 4	105	139	562	637
5.	Dokumen 5	25	28	128	145
6.	Dokumen 6	6	10	20	32
7.	Dokumen 7	48	79	554	631
8.	Dokumen 8	38	49	397	432
9.	Dokumen 9	0	0	4	6
10.	Dokumen 10	61	74	443	466

Tabel 6 menunjukkan jumlah perbedaan kata yang dimiringkan dengan *jaro winkler distance* dan pemiringan kata asing dengan *fuzzy string matching.* Pada tabel 6 menggambarkan bahwasanya pemiringan kata asing dengan *fuzzy string matching*, mengakibatkan kata yang dimiringkan menjadi lebih banyak. Dokumen satu, uji data 1 pemiringan dengan *jaro winkler* mengakibatkan 74 kata tercetak miring dan pemiringan kata dengan *fuzzy string matching* mengakibatkan 85 kata tercetak miring. Sedangkan, kondisi kata di dokumen satu pada uji data 2 ketika dimiringkan dengan *jaro winkler distance* terdapat 395 kata yang dimiringkan dan 423 kata yang dimiringkan dengan *fuzzy string* Begitu seterusnya hingga dokumen 10.

Tabel 7. Hasil Perhitungan Akurasi Uji Data 1

No.	Dokumen	Pemiringan dengan *Jaro Winkler Distance*				Pemiringan dengan *Fuzzy String Matching*			
		TP	TN	Jumlah kata pada Dokumen	Hasil	TP	TN	Jumlah kata pada Dokumen	Hasil
1.	Dokumen 1	74	1973	2391	85,61%	84	1972	2391	85,98%
2.	Dokumen 2	146	5328	6466	84,65%	173	5328	6466	85,07%
3.	Dokumen 3	76	2863	3144	93,47%	80	2863	3144	93,60%
4.	Dokumen 4	106	2425	3067	82,52%	139	2425	3067	83,59%
5.	Dokumen 5	22	3999	4140	97,12%	25	3999	4140	97,19%
6.	Dokumen 6	6	887	912	97,91%	9	886	912	98,13%
7.	Dokumen 7	48	1842	2452	77,07%	79	1842	2452	78,34%
8.	Dokumen 8	38	2775	3212	87,57%	49	2775	3212	87,92%
9.	Dokumen 9	0	1130	1130	100%	0	1130	1130	100%
10.	Dokumen 10	61	117	571	31,37%	74	117	571	33,45%

Tabel 7 merupakan tabel untuk menghitung akurasi pada dokumen saat uji data 1. Data yang dibutuhkan untuk mengetahui akurasi pada setiap dokumen adalah data nilai TP, TN dan jumlah data kata pada dokumen. TP merupakan nilai *true postive*, yaitu jumlah kata asing yang berhasil dimiringkan, TN merupakan nilai *true negative* yaitu nilai dari jumlah kata bukan asing yang berhasil tidak dimiringkan, jumlah kata pada dokumen dapat dilihat pada tabel 1. Nilai akurasi dapat dilihat pada kolom hasil yang didapat dari perhitungan *confusion matrix* yaitu menjumlahkan nilai TP dan TN terlebih dahulu kemudian dibagi dengan jumlah kata pada dokumen dan dikalikan dengan 100%. Rata-rata nilai akurasi pengujian didapat dari menumlahkan semua hasil akurasi setiap dokumen, kemudian dibagi dengan jumlah dokumen yaitu 10. Adapun rata-rata akurasi untuk pemiringan dengan *jaro winkler distance* adalah :

$$Rata-rata_{1} = \frac{85{,}61+84{,}65+93{,}47+82{,}52+97{,}12+97{,}91+77{,}07+87{,}57+100+31{,}37}{10} x100\% = 83{,}73\%$$

Sedangkan, rata-rata akurasi untuk pemiringan dengan *fuzzy string matching* adalah :

$$Rata-rata_{2} = \frac{85{,}98+85{,}07+93{,}60+83{,}59+97{,}19+98{,}13+78{,}34+87{,}92+100+33{,}45}{10} x100\% = 84{,}33\%$$

Berdasarkan nilai rata-rata satu dan dua dapat disimpulkan pemiringan kata dengan *fuzzy string matching* lebih besar 0,6%. Waktu pemrosessan dengan *fuzzy string matching* cenderung lebih cepat dan lebih banyak kata yang mampu dimiringkan.

Tabel 8. Hasil Perhitungan Akurasi Uji Data 2

No.	Dokumen	Pemiringan dengan *Jaro Winkler Distance*				Pemiringan dengan *Fuzzy String Matching*			
		TP	**TN**	**Jumlah kata pada Dokumen**	**Hasil**	**TP**	**TN**	**Jumlah kata pada Dokumen**	**Hasil**
1.	Dokumen 1	395	1973	2391	99,03%	413	1963	2391	99,37%
2.	Dokumen 2	1045	5328	6466	98,56%	1108	5290	6466	98,94%
3.	Dokumen 3	259	2863	3144	99,30%	280	2855	3144	99,71%
4.	Dokumen 4	560	2432	3067	97,55%	617	2405	3067	98,53%
5.	Dokumen 5	128	3999	4140	99,68%	141	3995	4140	99,90%
6.	Dokumen 6	20	887	912	99,45%	25	880	912	99,23%
7.	Dokumen 7	554	1842	2452	97,71%	593	1804	2452	97,75%
8.	Dokumen 8	397	2775	3212	98,75%	429	2772	3212	99,65%
9.	Dokumen 9	0	1126	1130	99,64%	0	1124	1130	99,46%
10.	Dokumen 10	443	117	571	98,07%	446	117	571	98,59%

Tabel 8 merupakan nilai akurasi dari uji data 2, sama halnya dengan cara pehitungan akurasi pada uji data 1 yaitu masih menggunakan perhitungan *confusion matrix.* Setelah menemukan nilai akurasi di setiap dokumen, langkah selanjutnya adalah menghitung nilai akurasi. Adapun rata-rata akurasi untuk pemiringan dengan *jaro winkler distance* adalah :

$$Rata-rata_{3} = \frac{99{,}03+98{,}56+99{,}30+97{,}55+99{,}68+99{,}45+97{,}71+98{,}75+99{,}64+98{,}07}{10} x100\% = 98{,}77\%$$

Sedangkan, rata-rata akurasi untuk pemiringan dengan *fuzzy string matching* adalah :

$$Rata-rata_{4} = \frac{99{,}37+98{,}94+99{,}71+98{,}53+99{,}90+99{,}23+97{,}75+99{,}65+99{,}46+98{,}59}{10} x100\% = 99{,}11\%$$

Nilai rata-rata 3 dan nilai rata-rata 4 merupakan hasil pemiringan kata dengan *jaro winkler distance* dan pemiringan kata dengan *fuzzy string matching* yang diambil pada saat uji data 2. perbedaan kedua nilai rata-rata tersebut adalah 0,34% lebih akurat pada pemiringan kata dengan *fuzzy string matching*.

4. Kesimpulan dan Saran

Aplikasi semi-otomatisasi dapat berjalan dengan baik, *jaro winkler distance* mampu mendeteksi kata asing pada dokumen. Namun, penerapan algoritma *jaro winkler* masih memiliki kekurangan yaitu terdapat pada ketidak tepatan pencocokan kata karena langkah perhitungan algoritma yang hanya menghitung kecocokan kata dari sisi kiri kata. Akan lebih baik jika perhitungan juga berlaku dari kiri kata dan kanan kata. Serta, algoritma *jaro winkler* kurang bisa memberikan toleransi nilai pada kata yang bermakna sama, seperti kata pada computer dengan

computers. Adanya penambahan *fuzzy string matching* pada penelitian ini, mampu memberikan nilai toleransi yang belum bisa diberikan oleh jaro winkler distance. Sehingga, dengan adanya penambahan *fuzzy string matching* pada pemiringan kata dengan algoritma j*aro winkler distance* dapat menghemat kata asing yang terdapat di database. Pemiringan kata dengan tambahan *fuzzy string matching* memerlukan waktu yang lebih singkat dari pada hanya dengan *jaro winkler distance.* Adanya *fuzzy string matching* juga menyebabkan nilai error karena ada beberapa kata yang bukan asing namun dicetak miring oleh aplikasi. Selama proses pencatatan waktu performa dan daya komputer akan mempengaruhi hasil waktu pemrosesan.
Saran yang dapat dilakukan untuk penelitan berikutnya adalah perbaikan metode atau cara untuk mengukur waktu pemrosesan karenanya selama penelitian ini berlansung nilai waktu pemrosesan baik dengan *jaro winkler* ataupun dengan penambahan *fuzzy string matching* seringkali berubah-ubah nilainya. Selain itu, perlu juga untuk mencoba algoritma pencocokan kata yang lain selain *jaro winkle distance*. Saran pada bidang komputasi agar dikembangkan mengenai pemrosesan kata dalam jumlah banyak menjadi proses komputasi yang lebih ringan.

[1] Daftar Pustaka

[2] Sugiyo, "Pengaruh Motivasi Belajar dan Penguasaan Kosakata Terhadap Kemampuan Menulis Narasi Siswa Kelas VIII SMP Mater Dei Pamulang Kota Tangerang Selatan", in *Jurnal Sasindo Unpam.* 3(2). 72-86, 2019, http://eprints.unpam.ac.id/id/eprint/1549

[3] Friendly, "Perbaikan Metode *Jaro–Winkler Distance* untuk *Approximate String Search* Menggunakan Data Terindeks Aplikasi *Multi User*" in Jurnal Teknovasi, Vol.04, No.2, 2017, 59-69. ISSN:2540-8389

[4] Putri, D.Z., Puspitaningrum, D. and Setiawan. Y, "Konversi Citra Kartu Nama Ke Teks Menggunakan Teknik OCR dan Jaro-Winkler *Distance" in* Jurnal TEKNOINFO, Vol. 12, No.1, 2018, 1-6. ISSN 1693-0010

[5] Prasetyo, A., Baihaqi, W. M.and Had. I. S., "Algoritma Jaro-Winkler Distance: Fitur Autocorrect Dan Spelling Suggestion Pada Penulisan Naskah Bahasa Indonesia Di Bms Tv", in Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), Vol. 5, No. 4, September 2018, hlm. 435-444. p-ISSN: 2355-7699. DOI: 10.25126/jtiik.201854780.

[6] Taufik, I. Aishia,I.D. and Jumadi.J, "Implementasi *Fuzzy Search* Untuk Pendeteksi Kata Asing Pada Dokumen *Microsoft Word." in Jurnal Teknik Informatika.* 10(1). 1-8, 2017, doi:10.15408/jti.v10i1.6804

[7] Mulyatun, S., Utama, H. and Mustopa A, "Pendekatan Natural language pada Aplikasi Cahtbot sebagai Alat Bantu Customer Service", in JOISM : Jurnal Of Information System Management Vol. 3, No. 1. e-ISSN : 2715-3088, 2021.

[8] Sihotang, M. T., Jaya, I., Hizriadi, A., and Hardi S. M, "Answering Islamic Questions with a Chatbot using Fuzzy String-Matching Algorithm" in .Journal of Physics: Conference Series, 2020 012007. doi:10.1088/1742-6596/1566/1/012007

[9] Frando, J., Ruslianto, I. and Hidayati, R, "Penerapan Jaro Winkler Distance dalam Aplikasi Pengoreksi Kesalahan Penulisan Bahasa Indonesia Berbasis Web" in *Jurnal Komputer dan Aplikasi.* 7(3). 44-53, 2019.

[10] Pinajeng, I, K, T, P., Sukarsa, I, M. and Putra, I, M, S, "Perbaikan Kata pada Sistem Chatbot dengan Metode Jaro Winkler" in JITTER- Jurnal Ilmiah Teknologi dan Komputer Vol. 1, No. 2, 2020.

[11] Tinaliah and Elizabeth, T, "Perbandingan Hasil Deteksi Plagiarisme Dokumen dengan Metode Jaro-Winkler Distance dan Metode Latent Semantic Analysis" in Jurnal Teknologi dan Sistem Komputer 6(1):7, 2018, DOI:10.14710/jtsiskom.6.1.2018.7-12.

[12] Kusumadewi, S., Hartati, S., Harjoko, A., Wardoyo, R "Fuzzy Multi-Atribute Decesion Makin (Fuzzy MADM). Yogyakarta : Graha Ilmu, 2006.

[13] Gunawan and Kirman, "Implementasi Algoritma Turbo Boyer Moore untuk Pencarian Data pada Transaksi Keuangan Duta Ponecell Sawah Lebar", in Jurnal Media Infotama Vol.15 No. 1, 2019.

[14] Anisah, S., Yulianto, T. and Faisol, F, "Perbandingan Fuzzy Sugeno dan Fuzzy Mamdani Pada Analisis Minat Masyarakat Terhadap Produk Air Minum Dalam Kemasan Lokal dan Nasional di Madura", in Zeta - Math Journal, 6(1), 29-37, 2021, https://doi.org/10.31102/zeta.2021.6.1.29-37

[15] Khatami, S, "Comparison and Improvement of Basic String Metrics for Surname Matching", in *Life Science Journal.* X(5). pp.128-32, 2013

[16] Oxford. (2020). Oxford Reference[online]. Available : https://www.oxfordreference.com/view/10.1093/oi/authority.20110803104449628.