ISSN : 1411-1624
e-ISSN: 2621-8089

APPROXIMATE STRING MATCHING UNTUK PENCARIAN KATA DALAM KAMUS BAHASA INDONESIA MENGGUNAKAN *ALGORITMA JARO WINK*LER

Vivi Sahfitri[1], Ibnu Batutah Zarizal [2]
Universitas Bina Darma [1,2]
Jalan Jenderal Ahmad Yani No.3 Palembang
Sur-el : vivi_sahfitri@binadarma.ac.id[1], ib_zarizal@gmail.com[2]

Abstract : *Language is the main thing in order to communicate with others both verbally and in writing. In language there is a meaning of words or sentences that can be accepted and understood if delivered with good and correct grammar. Dictionary is a tool used to know the meaning of a word. Currently, dictionaries are no longer conventional but are digital-based so that they can be accessed freely. This research aims to build a word search application in the dictionary Indonesian by applying the approximate String Matching method in word search on a Web-based Indonesian dictionary using jaro Winkler's algorithm. The research will also provide information about the time required by the system in string search, word suggestions, word meanings, sample sentences and images. Computational time testing results showed that jaro winkler's algorithm had a very effective quadratic runtime complexity on short strings. In addition, a lot of data in the database affects the search time but provides a word that is closer to the word searched. Functional testing to determine the performance of the software system shows that all parts of the system can run well according to the desired purpose when building the system.*

Keywords: *Aproximate String Matching, Word Searching, Indonesia Dictionary, Jaro Winkler Algorithm*

Abstrak : Bahasa merupakan kebutuhan utama agar dapat berkomunikasi dengan orang lain baik secara lisan maupun secara tertulis. Didalam bahasa terdapat makna kata atau kalimat yang dapat diterima dan dipahami jika disampaikan dengan tata bahasa yang baik dan benar. Kamus merupakan alat yang digunakan untuk mengetahui makna dari sebuah kata. Saat ini kamus tidak lagi bersifat konvensional akan tetapi sudah berbasis digital sehingga dapat di akses secara leluasa. Penelitian ini bertujan untuk membangun sebuah aplikasi pencarian kata dalam kamus bahasa Indonesia menerapkan metode approximate String Matching berbasis Web mengggunakan algoritma Jaro Winkler. Penelitian ini juga akan memberikan informasi tentang waktu yang diperlukan oleh sistem dalam pencarian string, saran kata, makna kata, contoh kalimat dan gambar. Hasil pengujian waktu komputasi menunjukan bahwa algoritma jaro winkler memiliki kompleksitas waktu quadratic yang sangat efektif pada string yang pendek. Selain itu banyak jumlah data didalam Database mempengaruhi waktu pencarian namun memberikan kata yang lebih mendekati kata yang dicari. Pengujian secara fungsional untuk mengetahu kinerja system perangkat lunak menunjukkan bahwa semua bagian system dapat berjalan dengan baik sesuai dengan tujuan yang diinginkan ketika membangin system tersebut.

Kata kunci*: Aproximate String Matching, Pencarian Kata, Kamus Bahasa Indonesia, Algoritma Jaro Winkler*

1. PENDAHULUAN

Bahasa Merupakan alat komunikasi yang sangat penting terutama dalam membangun interaksi dengan orang lain. Bahasa merupakan alat komunikasi lingual manusia baik secara lisan maupun tulisan [1]. Penggunaan tata bahasa yang baik dan benar memberikan kemudahan bagi lawan bicara untuk memahami makna kalimat yang diucapkan. Indonesia

ISSN : 1411-1624
e-ISSN: 2621-8089

merupakan Negara besar yang memiliki banyak pulau sehingga menjadikan Indonesia memiliki keanekaragaman salah satunya adalah Bahasa. Setiap pulau yang ada di Indonesia memiliki bahasa daerah masing-masing. Namun hal tersebut tidak menjadi masalah, karena Indonesia memiliki bahasa persatuan yaitu bahasa Indonesia. Untuk memudahkan dalam memahami makna kata dalam bahasa Indonesia, kamus merupakan alat bantu yang paling sering digunakan. Secara umum kamus menjadi sebuah rujukan yang digunakan untuk mengetahui makna atau arti sebuah kata [2]. Saat ini, kamus lebih mudah digunakan, karena dapat diakses secara *online* dengan menggunakan *web browser* yang terhubung dengan jaringan internet ataupun secara *offlin*e dengan menginstal aplikasi pada komputer maupun *smartphone*. Pada saat menggunakan kamus digital yang tersedia, pencarian kata pada kamus tersebut seringkali terjadi kesalahan pengetikan pada saat mencari kata yang diperlukan. Kesalahan kata pada saat pencarian dapat berpengaruh pada informasi yang didapat oleh pengguna. Pengguna akan mendapatkan informasi yang tidak tepat bahkan tidak mendapatkan informasi yang diinginkan.

Terdapat berbagai macam metode dan algoritma yang bisa diterapkan untuk membangun sebuah sistem yang dapat digunakan dalam pencarian kata dan memberikan saran kata yang paling mendekati sehingga informasi yang diinginkan dapat diperoleh walaupun terjadi kesalahan pada saat pengetikan kata yang dicari. Penggunaan algoritma dalam penyelesaian permasalahan sistem pencarian merupakan metode efektif karena dilakukan secara berurutan dan sistematis menggunakan bahas yang logis untuk memecahkan permasalaha [3]. Sistem pencarian seperti ini dapat membuat koreksi pada penambahan kata yang dicari dengan membandingkan *string* yang satu dengan string yang lain. Pembandingan *string* secara garis besar dibedakan mejadi dua yaitu; *exact String matching* merupakan pembandingan string secara tepat dengan susunan karakter dalam string yang dibandingkan dan *inexact string matching* merupakan pembandingan string dimana string yang dicocokan memiliki kemiripan tetapi keduanya memiliki susunan karakter yang berbeda [4] [5]. Metode yang dapat digunakan dalam koreksi objek string adalah *Approximate String Matching* dengan algoritma *Jaro Winkler*. Metode *Approximate String Matching* merupakan metode yang digunakan untuk mencocokan string dengan kemiripan didasarkan pada segi penulisannya dengan membandingkan dua buah string didasarkan tingkat kemiripan yang ditentukan dengan jauh tidaknya beda penulisan dua string tersebut [6]. Operasi sederhana yang dilakukan pada metode *Approximate String Matching* ini adalah penyisipan, Penghapusan, Substitusi dan transposisi [7]. Metode *Approximate String Matching* telah diuji dengan lima algoritma yang berbeda, yaitu: *Levenshtein, Jaro Winkler, Monge Elkan, bi-Gram* dan *Jaccrad*. Hasil dari pengujian menunjukkan bahwa *approximate string matching* memiliki tingkat akurasi, kinerja, pendekatan perkiraan, serta nilai MAP *(Mean Average Precission) metric* yang cukup baik [8].

ISSN : 1411-1624
e-ISSN: 2621-8089

Penelitian ini menerapkan Algoritma *Jaro Winkler*. algoritma *Jaro Winkler* merupakan varian dari Jaro *Distance Metric*, yaitu sebuah algoritma untuk mengukur kesamaan antara dua string. Biasanya Algoritma ini digunakan didalam pendeteksian duplikat [9]. Semakin tinggi *jaro winkler distance* untuk dua string maka semakin mirip dengan string tersebut. Nilai normal dari *Jaro Winkler distance* adalah nol yang menandakan tidak ada kesamaan dan satu yang menunjukkan adanya kesamaan [10] [11]. Algoritma Jaro Winkler Distance memiliki kompleksitas waktu *Quadratic Runtime Complexity* yang sangat efektif pada string pendek dan dapat bekerja lebih cepat dari algoritma *edit distan*ce.

Penelitian yang berkaitan dengan *approximate string matching* dan algoritma jaro Winkler sudah banyak dilakukan sebelumnya. Penelitian yang dilakukan oleh Friendly yang membahas tentang perbaikan metode Jaro Winkler Distance untuk *Approximate String Search* menggunakan data terindeks aplikasi multi user [12]. Penelitian yang dilakukan [13] menyimpulkan bahwa algoritma Jaro Winkler memiliki kompleksitas waktu *quadratic, runtime dan complexity* yang sangat efektif diimplemtasikan pada string pendek dan lebih cepat mendapatkan hasil dibandingkan dengan algoritma *Levenshtein Distance*. Menurut [14] algoritma Levensthein Distance dapat memberikan hasil yang baik untuk mengatasi masalah kesalahan ejaan. Hasil Penelitian [15] menemukan bahwa dengan urutan kata yang sama pada aplikasi yang mengimplementasikan algoritma jaro winkler dapat dengan baik mendeteksi plagiarism dalam suatu karya ilmiah. Penelitian [16] menerapkan *string matching* untuk pencarian berita utama pada portal berita berbasia android.

Penelitian ini bertujuan untuk mengimplementasikan metode *approximate String Matching* dalam pencarian kata pada kamus Bahasa Indonesia berbasis Web mengggunakan algoritma Jaro Winkler. Penelitian ini juga akan memberikan informasi tentang waktu yang diperlukan oleh sistem dalam pencarian string, saran kata, makna kata, contoh kalimat dan gambar

2. METODOLOGI PENELITIAN

2.1. Metode *Approximate String Matching*

Metode *Approximate string matching* merupakan salah satu algoritma yang betujuan untuk melakukan pencocokan atau kedekatan pola kata tertentu dalam sebuah kalimat atau teks panjang. Pencocokan kalimat ini didasarkan pada kemiripan penulisan string yang akan dibandingkan untuk mengetahui jauh tidaknya perbedaan penulisan dua string tersebut [17]. *Approximate string matching* merupakan metode string matching yang mencocokan string didasarkan pada kemiripan secara tekstual atau dari segi penulisan string yang dapat meliputi banyaknya karakter yang digunakan dan Susunan karakter dalam dokumen. Sebagai contoh pada penulisan kata *Pandai* dan *pandei* yang memiliki jumlah string yang sama tetapi terdapat karakter yang berbeda. Selain dari sisi tekstual yang menggunakan *approximate string matching*, pencocokan string berdasarkan

ISSN : 1411-1624
e-ISSN: 2621-8089

ucapan dapat menggunakan metoe *phonetic string matching*. Sebagai contoh pengucapan kata *Obat* dan *Obad* secara tekstual berbeda akan tetapi memiliki pengucapan hampir sama sehingga dua string tersebut dianggap cocok.

2.2. Algoritma Jaro Winkler

Algoritma Jaro Winkler merupakan metode yang digunakan untuk mengukur kesamaan string yang ada pada dua kata. Algortima ini sering digunakan untuk pencarian kata pada kalimat yang memiliki kecocokan untuk menghindari plagiarism. Kriteria dasar dari algoritma Jaro Winkler ini adalah menghitung panjang string, menemukan jumlah karakter yang sama dalam dua string dan menemukan jumlah transposisi [13].

Persamaan yang digunakan pada algoritma Jaro Winkler untuk menghitung panjang string adalah:

$$dj = \frac{1}{3}\left(\frac{c}{|s_1|} + \frac{c}{|s_2|} + \frac{c-t}{c}\right) \qquad (1)$$

Dimana:

c = Jumlah karakter yang sama
$|s_1|$ = Length karakter string ke-1
$|s_2|$ = Length Karakter string ke- 2
t = Jumlah Transposisi

Sedangkan untuk menghitung jarak yang dibenarkan untuk dua buah string menggunakan persamaan (2) berikut :

$$Jarak\ teoritis = \left(\frac{max\ (|s_1|,|s_2|)}{2}\right) - 1 \qquad (2)$$

Untuk membandingkan jarak String 1 dengan string 2 maka untuk mengetahui nilai Jaro Winkler distance menggunakan persamaan (3) berikut:

$$dw = dj + (lp(1 - dj)) \qquad (3)$$

Dimana:

dj = Jaro distance dari string 1(s1) dan String 2 (s2)
l = merupakan panjang prefix umum di awal string nilai maksimalnya 4 karakter
P = Konstanta *scaling factor*. Nilai Konstanta menurut winkler adalah p = 0,1

2.3. Tahapan penelitian

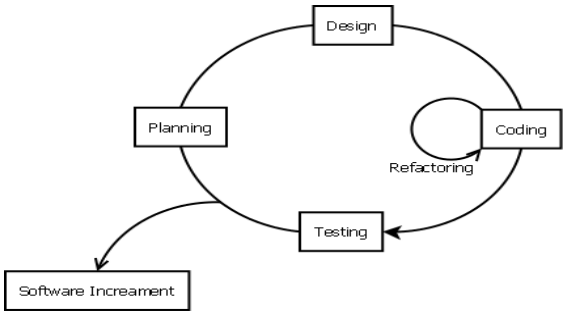
Tahapan penelitian di awali dengan mengumpulkan data melalui kegiatan studi literature dan observasi yang berkaitan dengan penelitian terutama tentang penerapan Algoritma Jaro Winkler [18]. Tahapan pengumpulan data sangat penting sebagai pondasi dalam melaksanakan penelitian yang sangat dibutuhkan pada tahapan pengembangan sistem. Dengan data yang diperoleh dari kegiatan pengumpulan data dapat dibangun sistem yang tepat dan sesuai dengan kebutuhan pengguna serta dapat menyelesaikan masalah yang ada. Tahap penelitian ini meliputi tahapan pengumpulan data, tahapan pengembangan sistem dan tahapan pengujian sistem. Gambar 1 menunjukkan diagram alur dalam penelitian ini.

Gambar 1. Diagram Alur Penelitian.

ISSN : 1411-1624
e-ISSN: 2621-8089

2.4. Metode Pengembangan Sistem

Tahap Perancangan Sistem yang digunakan dalam penelitian ini menggunakan metode *Extreme Programming.* Metode *extreme Programming* merupakan salah satu metode pengembangan sistem yang bertujuan untuk menghasilkan perangkat lunak yang memiliki kualitas tinggi dan produktif. Tahapan pada metode *extreme programming* di mulai dari tahap perencanaan (*Planning),* Perancangan (*Design),* Pengkodean (*Coding),* dan Pengujian (*testing)* [19]. Gambar 2 menunjukkan tahapan metode *extreme programming*



Gambar 2. Metode *Extreme Programming*

Metode Pengembangan Sistem *Extreme Programming* terdiri dari 4 tahap yaitu Perencanaan, Perancangan, Pengkodean dan pengujian. Masing masing tahapan saling berkaitan dan saling mendukung untuk membangun sistem yang sesuai dengan tujuan yang ingin dicapai.

1. Perencanaan (*Planning),* merupakan tahap perencanaan untuk merencanakan konteks dan konten dari aplikasi, output yang dihasilkan, fitur atau menu yang akan dibuat pada aplikasi. Selain itu pada tahap ini juga dilakukan penentuan waktu dan biaya untuk melakukan pengembangan system.
2. Perancangan (*Design),* Merupakan tahap untuk merancang aplikasi yang akan dibangun. Perancangan ini dapat berupa *flowchart, Use case diagram, Activity Diagram* dan antar muka system (*User Interface)*
3. Pengkodean (*Coding),* merupakan tahapan untuk mengimplementasikan rancangan yang sudah dibuat kedalam suatu bahasa pemrograman tertentu untuk menghasilkan perangkat lunak yang siap digunakan. Metode *extreme programming* digunakan untuk membangun system berskala kecil sehingga untuk mengimplementasikan rancangan kedalam program tidak memerlukan banyak *programmer.*
4. Pengujian (*testing),* pada tahap ini dilakukan pengujian terhadap fitur yang ada pada aplikasi yang telah dibangun agar saat digunakan tidak terjadi *error*/kesalahan serta aplikasi yang dibuat sesuai dengan proses bisnis yang sudah dirancang sebelumnya.

3. HASIL DAN PEMBAHASAN

Hasil yang diperoleh dalam penelitian ini adalah rancangan sistem yang terdiri dari *flowchart system, Use case diagram, Activity diagram* dan *User interface system.*

3.1. Perencanaan (Planning)

Langkah awal dalam merancang dan membangun sistem *word searching* dengan mengimplementasikan metode *approximate string matching* menggunakan algoritma Jaro winkler adalah dengan melakukan analisis

ISSN : 1411-1624
e-ISSN: 2621-8089

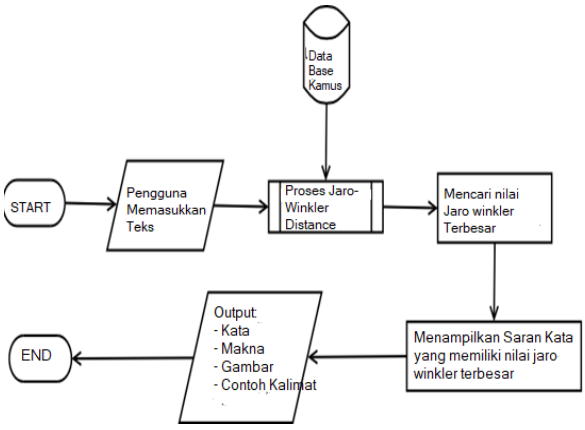
kebutuhan perangkat lunak sesuai dengan tahapan dalam metode pengembangan sistem yang digunakan yaitu *extreme programming*. Pada tahap analisis ini dilakukan untuk mengetahui kebutuhan perangkat lunak yang akan dibangun. Perencanaan yang dihasilkan akan dibuat dalam bentuk *flowchart* yang menggambarkan langkah langkah proses komputasi dari perangkat lunak [20], diagram-diagram yaitu *use case diagram* yang akan menggambarkan model fungsional dari pernagkat lunak yang akan dibangun, *Activity diagram* yang akan menggambarkan *workflow* proses bisnis dan urutan proses pada perangkat lunak yang dibangun [21], serta *user interface* yang akan menjadi antarmuka antara system dengan penggunanya [22].

3.2. Perancangan (design)

Tahap perancangan merupakan tahapan untuk merancang perangkat lunak yang akan dibangun dengan mengimplementasikan rencana yang sudah dibuat dari hasil analisis dalam rancangan *Flowchart System, Use Case Diagram, use case scenario, Activity Diagram,* dan *user interface.*

1. *Flowchart System*

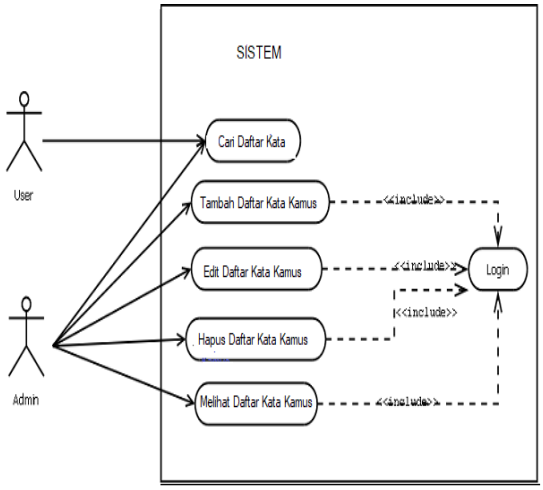
Flowchart yang dibuat merupakan langkah-langkah yang menggambarkan proses komputasi yang terjadi dalam system pencarian kata dalam kamus bahasa Indonesia menerapkan Metode *Approximate string matching* dengan algoritma *Jaro Winkler*[23].



Gambar 3. *Flowchat System* Pencarian Kata

2. *Use Case Diagram*

Perancangan *Use case Diagram* bertujuan untuk memudahkan pemberian penjelasan secara rinci yang berhubungan dengan interaksi antara *actor* atau pengguna dengan system yang akan dibangun. *Use Case Diagram* adalalah rancangan atau model prilaku (*Behaviour)* system yang akan dibangun [21]. Gambar 4 menunjukkan hubungan antara actor dengan system.



Gambar 4. *Use case Diagram*

1. Antarmuka Sistem

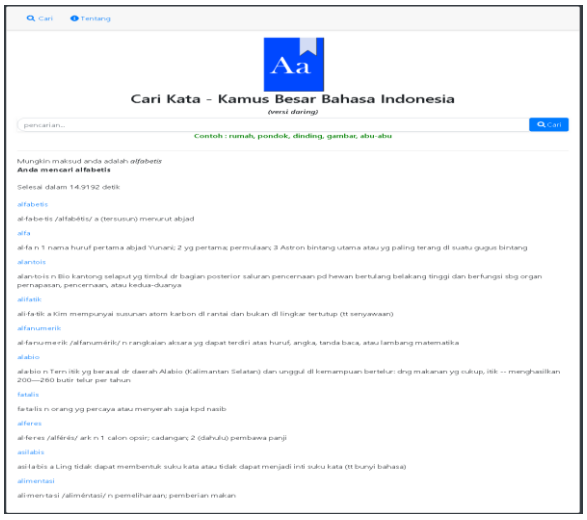
Antarmuka atau interface pada sebuah perangkat lunak yang dibangun merupakan

ISSN : 1411-1624
e-ISSN: 2621-8089

media penghubung antara pengguna (*User*) dengan perangkat Lunak [24]. Berikut antarmuka perangkat lunak yang telah dibangun.

Gambar 5. Menu *Home*

Menu utama merupakan tampilan awal perangkat lunak yang menampilkan form pencarian kata yang diinginkan. Pada halaman ini terdapat menu cari dan menu tentang yang memuat penjelasan tentang perangkat lunak ini.

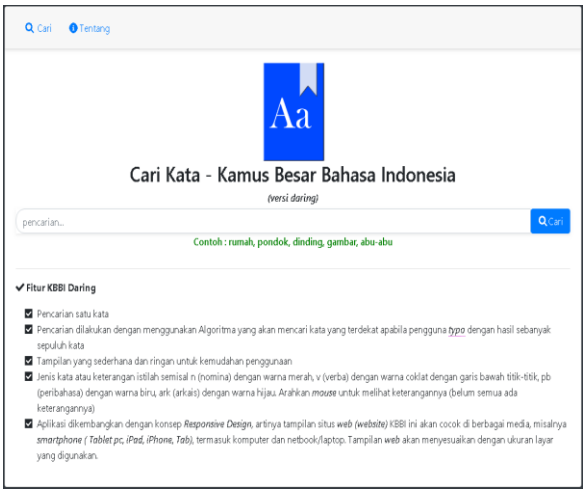


Gambar 6. Hasil Pencarian Kata

Halaman ini menampilkan hasil pencarian setelah pengguna memasukkan kata pada form pencarian dan menekan tombol cari pada *form* tersebut. Algoritma *Jaro-Winkler* yang diimplementasikan pada perangkat lunak akan menampilkan hasil pencarian yang mendekati kata yang dicari beserta makna dari kata tersebut serta rentang waktu pencarian sampai menampilkan hasil pencarian.

Gambar 7. Detail Pencarian Kata

Tampilan halaman detail pencarian kata akan menampilkan detail pencarian untuk satu kata yang dicari dengan menampilkan kata, arti dan gambar dari kata yang dipilih.



Gambar 8. Halaman Tentang

3.3. Pengujian Algoritma Jaro Winkler

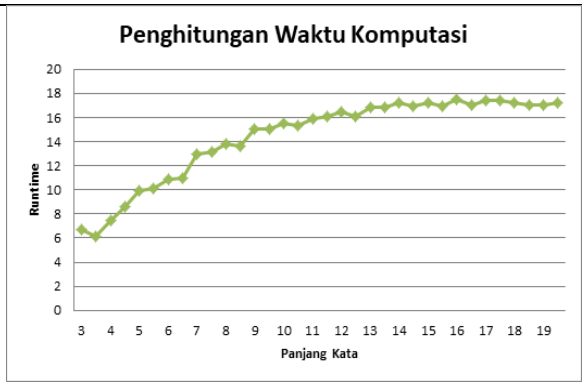
Pengujian dilakukan untuk mengetahui apakah system yang dibangun telah sesuai dengan rencangan yang sudah dibuat. Selain itu pengujian juga dilakukan untuk meminimalisir kesalahan *(error)* sehingga *output* yang dihasilkan sesuai dengan yang diinginkan tanpa ada kesalahan saat pengujian dilakukan. Tabel 1 menunjukkan hasil pengujian beberapa kata yang dicari dengan aplikasi pencarian kata dengan menggunakan Algoritma *Jaro - Winkler*.

ISSN : 1411-1624
e-ISSN: 2621-8089

Tabel 1. Pengujian kata dengan Algoritma *Jaro-Winkler*

No.	*Kata*		*Jumlah Kata*	*Waktu (detik)*	*Score*	*Index Ke-*
	Kata Tepat	*Kata Tidak Tepat*				
1	Abi	Avi	3	6.2321	0.8000000119209163	10
2	Uap	Uao	3	6.54785	0.8222222328183021	9
3	Baca	baza	4	7.492	0.8666666746139526	8
4	Calon	cakon	5	8.9421	0.8932333158497854	1
5	Zaman	jaman	5	9.1451	0.8966666746139471	28
6	Daerah	daersh	6	10.9083	0.9333333373069665	1
7	Wahana	wahama	6	10.9201	0.9333333373069763	2
8	wilayah	wilahah	7	13.159	0.9095237851142883	3
9	fakultas	fakulyas	8	13.8402	0.9166666865346675	1
10	yudisium	yusisium	8	13.6207	0.8952381014823914	1
11	gelembung	gelemnung	9	15.0572	0.9421296119689941	1
12	wisudawan	wisucawan	9	15.0386	0.9555555582046509	1
13	halusinasi	haluainasi	10	15.3296	0.9377778172492981	1
14	verifikasi	veripikasi	10	15.4774	0.9599999785423279	1
15	intelektual	interektual	11	15.9225	0.9036363363265991	11
16	ultraviolet	ultrapiolet	11	16.0414	0.9696969985961914	1
17	jukstaposisi	juxstaposisi	12	16.4565	0.9555555582046509	1
18	transformasi	transpormasi	12	16.0997	0.9722222089767456	2
19	kontroversial	kontrofersial	13	16.8167	0.9794871807098389	1
20	superkomputer	supercomputer	13	16.803	0.9743589758872986	1
21	leukositometer	liukositometer	14	17.1725	0.8994505405426025	1
22	telekomunikasi	telecomunikasi	14	16.975	0.9252747297286987	1
23	multirasialisme	multilasialisme	15	17.2229	0.9480158686637878	1
24	sinemikrografik	sinemakrografik	15	16.9421	0.9599206447601318	1
25	neoimpresionisme	meoimpresionisme	16	17.4645	0.9138888716697693	1
26	interaksionistik	interaksianistik	16	17.0151	0.9833333492279053	1
27	piezoelektrisitas	piesoelektrisitas	17	17.3697	0.8777573108673096	2
28	fotokonduktivitas	potokonduktivitas	17	17.4501	0.9607843160629272	1
29	retradisionalisasi	ritradisionalisasi	18	17.1741	0.8254901766777039	1
30	interdepartemental	interdefartemental	18	17.0067	0.9851852059364319	1
31	tetrahidrokanabinol	tetrahigrokanabinol	19	17.0352	0.9859648942947388	1
32	elektroensefalogram	elektroinsefalogram	19	17.2481	0.9748538136482239	1

Hasil pengujian algoritma *Jaro – Winkler* dalam melakukan pencarian kata yang diminta yang diukur berdasakan waktu yang diperlukan untuk menemukan kata yang dimaksud. Hasil pengujian direpresentasikan dalam grafik dapat dilihat pada gambar 9 berikut.



Gambar 9. Grafik perhitungan waktu komputasi pencarian kata

ISSN : 1411-1624
e-ISSN: 2621-8089

Pada Algoritma Jaro-Winkler yang diimplementasikan pada pencarian string, semakin panjang suatu string maka waktu yang dibutuhkan untuk pencarian akan semakin tinggi [13]. Algoritma Jaro-winkler sangat efektif jika diimplementasikan pada string pendek karena algoritma ini memiliki komplesitas waktu *quadratic runtime complexity* yang memungkinan memperoleh hasil pencarian yang lebih cepat.

3.4. Pengujian Perangkat Lunak

Pengujian pada sistem pencarian kata pada kamus Bahasa Indonesia dengan menerapkan metode *Approximate String Matching* berbasis Web dilakukan dengan menggunakan *Black Box Testing. Black Box Testing* merupakan salah satu jenis metode pengujian untuk perangkat lunak yang belum diketahui kinerja inetrnalnya [25]. Berikut pengujian yang dilakukan pada sistem Perangkat Lunak yang dibangun.

1. Pengujian Login system, dilakukan untuk melihat proses verifikasi *username dan Password* yang dilakukan oleh admin.

Tabel 2. Pengujian Login Sistem

Data Masukan	*Yang Diharapkan*	*Hasil Pengamatan*	*Kesimpulan*
Username: admin, *password*: admin	admin tercantum pada *form username*, admin tercantum pada *form password*	admin tercantum pada *form username*, ***** tercantum pada *form password*	Diterima
Klik tombol Login	Login admin dicari di table admin, masuk ke halaman utama admin.	Tombol login dapat berfungsi. Sesuai yang diharapkan.	Diterima

2. Pengujian Halaman Tambah Kata, dilakukan untuk mengetahui proses penambahan kata kedalam sistem sudah berjalan dengan baik atau tidak.

Tabel 3. Pengujian Halaman Tambah Kata

Data Masukan	*Yang Diharapkan*	*Hasil Pengamatan*	*Kesimpulan*
Klik tombol Tambah Kata	Menampilkan form tambah kata	Tampil form tambah kata	Diterima
Kata : citra; arti : rupa, gambar, gambaran; kalimat : Mereka memiliki alat pengamat citra; gambar;	Kata tercantunm dalam form, arti tercantum dalam form, kalimat tercantum dalam form, nama gambar tercantum dalam form	Dapat mengisi kata, arti, kalimat gambar. Sesuai yang diharapkan.	Diterima
Klik Simpan	Data tersimpan pada *database* dan ditambilkan didalam *tablegrid*.	Data tersimpan didalam database dan tampil di *tablegrid*.	Diterima

3. Pengujian Halaman Edit Kata, dilakukan untuk mengetahui proses pengeditan kata pada sistem sudah berjalan dengan baik atau tidak.

Tabel 4. Pengujian Halaman edit Kata

Data Masukan	*Yang Diharapkan*	*Hasil Pengamatan*	*Kesimpulan*
Klik tombol Edit	Menampilkan kata yang akan diubah di halaman edit kata	Menampilkan kata yang akan diubah. Sesuai yang diharapkan.	Diterima
Klik tombol Simpan	Data perubahan tersimpan didalam table kbbi. Kembali ke halaman data kamus.	Data perubahan tersimpan didalam table kbbi. Kembali ke halaman data kamus. Sesuai yang diharapkan.	Diterima
Klik Kembali	Data perubahan tidak tersimpan di table kbbi. Kembal ke halaman data kamus.	Data perubahan tidak tersimpan di table kbbi. Kembal ke halaman data kamus.	Diterima

ISSN : 1411-1624
e-ISSN: 2621-8089

4. Pengujian Hapus Kata, dilakukan untuk mengetahui proses penghapusan kata pada sistem sudah berjalan dengan baik atau tidak.

Tabel 5. Pengujian Hapus Kata

Data Masukan	*Yang Diharapkan*	*Hasil Pengamatan*	*Kesimpulan*
Klik tombol Hapus	Menampilkan notifikasi beserta kata yang akan dihapus.	Menampilkan notifikasi beserta kata yang akan dihapus. Sesuai yang diharapkan.	Diterima
Klik tombol Botol	Notifikasi hilang dan data tidak terhapus.	Notifikasi hilang dan data tidak terhapus. Sesuai yang diharapkan.	Diterima

5. Pengujian Pencarian Kata, dilakukan untuk mengetahui proses pencarian kata pada sistem sudah brjalan dengan baik atau tidak.

Tabel 6. Pengujian Pencarian Kata

Data Masukan	*Yang Diharapkan*	*Hasil Pengamatan*	*Kesimpulan*
Kata adaptasi	Kata tercantum pada *form* pencarian	Kata tercantum pada *form* pencarian. Sesuai yang diharapkan.	Diterima
Klik tombol Cari	Kata yang dicari ditemukan	Kata yang dicari ditemukan. Sesuai yang diharapkan.	Diterima

6. Pengujian Pencarian Kata Algoritma *Jaro-Winkler,* dilakkan untuk mengetahui proses pencarian kata dengan menerapkan algoritma Jaro winkler pada sistem sudah berjalan dengan baik atau tidak.

Tabel 7. Pengujian Pencarian Kata Algoritma *Jaro-Winkler*

Data Masukan	*Yang Diharapkan*	*Pengamatan*	*Kesimpulan*
Kata : kosnultasi	Kata tercantum pada *form* pencarian	Kata tercantum pada *form* pencarian. Sesuai yang diharapkan.	Diterima
Klik tombol Cari	Memberikan saran kata yang dicari. Menampilkan rentang waktu pencarian.	Kata yang tepat ditemukan. Menampilkan rentang waktu pencarian. Sesuai yang diharapkan.	Diterima

4. KESIMPULAN

Berdasarakan tahapan penelitian yang telah dilakukan, kesimpulan dari peneltian ini adalah Aplikasi pencarian kata pada kamus Bahasa Indonesia dengan metode *Approximate String Matcing* yang menerpakan Algoritma *Jaro Winkler* berbasis Web dapat membantu dalam mengatasi permasalahan kesalahan pengetikan pada sebuah kata. Proses pengujian terhadap waktu komputasi memperoleh hasil bahwa algoritma *jaro Winkler* memiliki komplesitas waktu *quadratic runtime complexity* yang sangat efektif pada string yang pendek. Jumlah data yang ada pada basis data merupakan factor utama pada proses pencarian dengan *Algoritma Jaro Winkler* untuk mendapatkan kata yang sesuai. Semakin besar basis data yang digunakan maka semakin banyak waktu yang dibutuhkan dalam proses pencarian. Akan tetapi karakatr yang tidak tepat saat pencarian kata juga akan mempengaruhi waktu yang dibutuhkan dalam proses pencarian.Pengujian terhadap kinerja fungsional system dengan metode *Black Box Testing*, dapat diketahui bahwa semua kinerja fungsional system perangkat lunak dapat bekerja dengan baik.

Penelitian ini belum melakukan uji kepuasan pengguna dari system pencarian kata yang dibangun, untuk mengetahui masukkan dari pengguna akhir system ini. Sehingga untuk penelitian berikutnya, dapat dilakukan pengukuruan terhadap indeks kepuasan

ISSN : 1411-1624
e-ISSN: 2621-8089

pengguna. Selain itu penerapan algoritma *jaro Winkler* dalam proses pencarian kata dapat dikombinasikan dengan algortima lainnya yang berkaitan dengan algoritam pencarian kesamaan teks (*text similarity)* antar dua kata.

DAFTAR PUSTAKA

[1] Adriyani, Ni Made Muni." Implementasi Algoritma Levenshtein Distance dan Metode Empiris Untuk Menampilkan Saran Perbaikan Kesalahan Pengetikan Dokumen Berbahasa Indonesia. *Jeliku (jurnal elektronik ilmu komputer udayana),* Vol.1 No.1, Agustus 2012.

[2] Ibrahim, Muhammad Yusuf, dan Nurgiyatna, "Kamus Lima Bahasa Dengan Metode Binary Search Dan Levenshtein Distance Berbasis Android"., 5 Agustus 2016. [Online] Tersedia: http://eprints.ums.ac.id/id/eprint/44974 [diakses: 10 desember 2021]

[3] Alfina, T., Santosa, B., & Barakbah, A. R., "Analisa Perbandingan Metode Hierarchical Clustering, K-means, dan Gabungan Keduanya dalam Cluster Data (Studi kasus: Problem Kerja Praktek Jurusan Teknik Industri ITS*)," JURNAL TEKNIK ITS*, Vol. 1 No. 1, pp. A521-A525, 2012.

[4] Sagita, V. & Prasetiyowati, M.I., "Studi Perbandingan Implementasi Algoritma Boyer-Moore, Turbo Boyer-Moore, dan Tuned Boyer-Moore dalam Pencarian String," *Ultimatics*, IV(1), pp.31-37, 2012.

[5] Syaroni, M., & Munir, R. "Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching) Dalam Bahasa Inggris," *Seminar Nasional Aplikasi Teknologi Informasi .2015*.

[6] Friendly, Friendly. "Perbaikan Metode Jaro–Winkler Distance Untuk Approximate String Search Menggunakan Data Terindeks Aplikasi Multi User." *Jurnal Teknovasi : Jurnal Teknik dan Inovasi* . Vol. 4, No.2, 2018.

[7] G. Navarro, "A guided Tour to Approximate String Matching," *ACM Computing Surveys*, Vol.33, No.1, pp.31-88, 2010.

[8] Braddley, M. O., Fachrurrozi, M., & Novi, Y., " Pengoreksian Ejaan Kata Berbahasa Indonesia Menggunakan Algoritma Levensthein Distance," *Prosiding Annual Research Seminar*, 3(1), 167–171, 2017.

[9] Rochmawati, Y., & Kusumaningrum, R, "Studi Perbandingan Algoritma Pencarian String dalam Metode Approximate String Matching untuk Identifikasi Kesalahan Pengetikan Teks," *Jurnal Buana Informatika*, Vol. 7, No. 2, pp.125–134, 2016.

[10] Kornain, A., Yansen, F., & Tinaliah, T, " Penerapan Algoritma Jaro-Winkler Distance untuk Sistem Pendeteksi Plagiarisme pada Dokumen Teks Berbahasa Indonesia," STMIK MDP, 2014.

[11] Khatami, S., " Comparison and Improvement of Basic String Metrics for Surname Matching," *Life Science Journal*) Vol. X No.5, pp.128-32, 2013.

[12] Friendly, "Perbaikan Metode Jaro-winkler Distance untuk Approximate String Search Menggunakan Data Terindeks Aplikasi Multi User," *Jurnal Teknovasi (Jurnal Teknik dan Inovasi)*, Vol. 04, No. 2, pp. 69, 2017.

[13] Yulianingsih, "Implementasi Algoritma Jaro Winler dan Levenstein Distance dalam Pencarian Data pada Database," *Jurnal String,* Vol. 02, No. 1, pp.18, Agustus 2017.

[14] Ir. Muhammad Aswin, Mt., Rachmania Nur Dwitiyastuti., Adharul Muttaqin, ST., MT. "Pengoreksi Kesalahan Ejaan Bahasa Indonesia Menggunakan Metode Levenshtein Distance." *Jurnal Mahasiswa Teknik Elektro Universitas Brawijaya*, vol. 1, no. 2, 2013.

[15] Kurniawati, A. Puspotodjati,S, Rahmat, S., " Implementasi Algoritma Jaro Winkler Distance untuk Membandigkan Kesamaan Dokumen Berbahasa Indonesia," 27 Februari 2014, [Online]. Tersedia : http://repository.gunadarma.ac.id/id/eprint/394. [Diakses: 20 januari 2022].

[16] Ernawati, Johar, A, Setiawan, S., " Implementasi String Matching Untuk Pencarian Berita Utama pada Portal Beritas Berbasis Android (Studi Kasus : Harian Rakyat Bengkulu), *Jurnal Pseudocode*, Vol. 1, No. 1, pp. 77. Februari 2019.

[17] Gurning, Ardi Isbad Amar, Zarnelly Zarnelly, dan Arabiatul Adawiyah. "Penerapan Fuzzy String Matching Pada

ISSN : 1411-1624
e-ISSN: 2621-8089

Aplikasi Pencarian Tugas Akhir Mahasiswa Jurusan Sistem Informasi Berbasis Web (Studi Kasus: Fakultas Sains dan Teknologi UIN Suska Riau)"., *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*. Vol. 2 No.1 pp. 54-59. Februari 2016

[18] Sugiyono. *Metode Penelitian Kuantitatif, Kualitatif, dan R&D.* Penerbit : Alfabeta. Bandung. 2017

[19] Suryantara, I G.N., *Merancang Aplikasi dengan Metodologi Extreme Programmings*. Gramedia , 2017.

[20] Indrajani. *Database Design (Case Study All in One)*. Jakarta: PT Elex Media Komputindo. 2015.

[21] A. S. Rosa and M. Shalahuddin, *Rekayasa perangkat lunak terstruktur dan berorientasi objek*. Bandung: Informatika, 2015.

[22] W.O Galitz., *The essential Guide to Unser Interface Design,* Canada: John Wiley & Sons. 2007

[23] Oktamal, F., Saptono, R., Sulistyo, M.E., " Jaro-Winkler Distance Dan Stemming Untuk Deteksi Dini Hama Dan Penyakit Padi," *SESINDO* 2015,

[24] Sahfitri, V. (2020). Perancangan sistem reservasi dan promosi hotel berbasis website. *J Inform*, *20*(1), 54-66.

[25] Salamah, U., & Khasanah, F. N. (2017). Pengujian Sistem Informasi Penjualan Undangan Pernikahan Online Berbasis Web Menggunakan Black Box Testing. *Information Management For Educators And Professionals : Journal Of Information Management*, Vol.2 No.1. pp. 35-46. Desember 2017.