

***JARO-WINKLER DISTANCE* UNTUK KOREKSI EJAAN KATA
MENGUNAKAN *MULTIPROCESSING* PADA SISTEM Pencarian
BERITA PARIWISATA**

SKRIPSI



Disusun oleh:

Muhammad Ali Yusuf

180411100135

Dosen Pembimbing 1 : Dr. Fika Hastarita Rachman, S.T., M.Eng.

Dosen Pembimbing 2 : Husni, S.Kom., M.Kom.

**PROGRAM STUDI INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA**

2023

HALAMAN JUDUL

***JARO-WINKLER DISTANCE* UNTUK KOREKSI EJAAN KATA
MENGUNAKAN *MULTIPROCESSING* PADA SISTEM Pencarian
BERITA PARIWISATA**

SKRIPSI

**Diajukan untuk Memenuhi Persyaratan Penyelesaian Studi Strata Satu (S1)
dan Memperoleh Gelar Sarjana Komputer (S.Kom.)
di Universitas Trunojoyo Madura**

**MUHAMMAD ALI YUSUF
(18.04.111.00135)**

**PROGRAM STUDI INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA**

2023

HALAMAN PENGESAHAN

***JARO-WINKLER DISTANCE* UNTUK KOREKSI EJAAN KATA
MENGUNAKAN *MULTIPROCESSING* PADA SISTEM Pencarian
BERITA PARIWISATA**

Oleh:

Nama : Muhammad Ali Yusuf

Nim : 180411100135

Disetujui oleh Tim Penguji Skripsi:

Tanggal Sidang:

17 Maret 2023

Dr. Fika Hastarita Rachman, S.T., M.Eng.

(Pembimbing I)

NIP. 19830305 200604 2 002

Husni, S.Kom., M.Kom.

(Pembimbing II)

NIP. 19770722 200312 1 006

Ari Kusumaningsih, S.T., M.T.

(Penguji I)

NIP. 19790222 200501 2 003

Dr. Bain Khusnul Khotimah, S.T., M.Kom.

(Penguji II)

NIP. 19800325 200312 2 002

Yudha Dwi Putra Negara, S.Kom., M.Kom.

(Penguji III)

NIP. 19890530 201903 1 012

Bangkalan, Maret 2023

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Yeni Kustivahningsih, S.Kom., M.Kom.

NIP. 19770921 200812 2 002

HALAMAN PERNYATAAN ORISINALITAS

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya dengan judul:

***JARO-WINKLER DISTANCE* UNTUK KOREKSI EJAAN KATA
MENGUNAKAN *MULTIPROCESSING* PADA SISTEM PENCARIAN
BERITA PARIWISATA**

1. Adalah asli, bukan merupakan karya pihak lain serta belum pernah diajukan untuk mendapatkan gelar akademik Sarjana Komputer baik di Universitas Trunojoyo Madura maupun di Perguruan Tinggi yang lain.
2. Tidak terdapat karya atau pendapat pihak lain yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis telah diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila dikemudian hari terbukti skripsi ini sebagian atau seluruhnya merupakan hasil plagiasi atau terdapat hal-hal yang tidak sesuai dengan pernyataan di atas, maka saya sanggup menerima sanksi akademis yang berlaku dengan segala akibat hukumnya sesuai peraturan Universitas Trunojoyo Madura dan atau peraturan perundangan-undangan yang berlaku.

Bangkalan, 13 Maret 2023

Yang Menyatakan,



Muhammad Ali Yusuf
NIM. 180411100135

KATA PENGANTAR

Segala puji dan syukur penulis ucapkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayahnya sehingga penulis dapat menyelesaikan skripsi dengan judul “*Jaro-Winkler Distance* untuk Koreksi Ejaan Kata Menggunakan *Multiprocessing* pada Sistem Pencarian Berita Pariwisata” sebagai salah satu syarat untuk menyelesaikan Studi Strata Satu (S1) di Program Studi Teknik Informatika Fakultas Teknik Universitas Trunojoyo Madura. Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini tidak terlepas dari dukungan, bimbingan, bantuan dari berbagai pihak yang sangat berarti sehingga skripsi ini dapat diselesaikan. Dengan ketulusan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Kedua orang tua dan keluarga yang tiada henti memberikan dukungan, kekuatan, doa, motivasi, nasihat-nasihat berharga, dan seluruh kebaikan yang selalu diberikan sehingga telah sampai pada tahap ini.
2. Ibu Dr. Fika Hastarita Rachman, S.T., M.Eng. dan Bapak Husni, S.Kom., M.Kom. selaku dosen pembimbing yang telah memberikan arahan dan motivasi kepada penulis sehingga dapat menyelesaikan skripsi dengan baik.
3. Ibu Ari Kusumaningsih, S.T., M.T. selaku Dosen Penguji I, Ibu Dr. Bain Khusnul Khotimah, S.T., M.Kom. selaku Dosen Penguji II, dan Bapak Yudha Dwi Putra Negara, S.Kom., M.Kom. selaku Dosen Penguji III yang telah banyak memberikan saran demi kemajuan dan penyempurnaan dalam pengerjaan skripsi ini.
4. Seluruh Bapak dan Ibu dosen staf pengajar di lingkungan Program Studi Teknik Informatika, Universitas Trunojoyo Madura yang telah memberikan ilmu dan pengetahuan kepada penulis selama menempuh studi.
5. Teman - teman yang selalu memberikan dukungan dan membantu penulis selama proses penulisan dan penyusunan laporan skripsi dari awal hingga akhir.
6. Seluruh pihak yang belum penulis sebutkan, terima kasih atas uluran tangan dan dorongan baik yang telah diberikan.

Dalam penulisan skripsi ini penulis menyadari masih jauh dari kata sempurna. Oleh karenanya, segala saran dan kritik yang membangun diharapkan dapat menyempurnakan skripsi ini serta dapat memberikan manfaat bagi penulis dan pembaca pada umumnya.

Bangkalan, 13 Maret 2023

Penulis

**JARO-WINKLER DISTANCE UNTUK KOREKSI EJAAN KATA
MENGUNAKAN *MULTIPROCESSING* PADA SISTEM Pencarian
BERITA PARIWISATA**

ABSTRAK

Perkembangan teknologi informasi saat ini mengubah pola hidup masyarakat salah satu di antaranya adalah dalam hal mencari informasi, termasuk berita pariwisata sebagai contohnya. Dalam memutuskan untuk liburan ke suatu tempat, wisatawan memerlukan beberapa referensi berita baik berupa rubrik atau artikel *online*. Wisatawan dapat mencari referensi tersebut melalui portal berita pariwisata dengan menggunakan mesin pencarian kemudian sistem akan memberikan umpan balik berdasarkan *keyword* yang diberikan. Namun dalam kenyataannya sering kali hasil yang diberikan sistem tidak sesuai dengan interpretasi yang dimaksud oleh *user*, beberapa faktor penyebabnya adalah terdapat kesalahan dalam memberikan *keyword* yang tidak termasuk ejaan baku atau terdapat kesalahan ketik (*typo*). Hal tersebut dapat ditangani dengan membangun suatu sistem yang mampu mengoreksi kesalahan pengetikan agar sesuai kaidah penulisan menggunakan algoritma *Jaro-Winkler Distance* yang dapat digunakan untuk menghitung jarak kemiripan antara dua *string*, kemudian dievaluasi dengan menerapkan *multiprocessing* untuk meningkatkan waktu eksekusi sistem. Hasil pengujian menunjukkan algoritma *Jaro-Winkler Distance* mampu mengoreksi dan memberikan saran kata pada kata yang kurang tepat dengan rata-rata akurasi pada semua jenis kesalahan sebesar 0.86. Selanjutnya pengujian terhadap *multiprocessing* menunjukkan bahwa model pemrosesan secara paralel mampu mempercepat waktu eksekusi pada setiap perangkat yang diujikan. Waktu eksekusi terpendek terdapat pada perangkat MacBook Pro baik pada proses secara serial maupun paralel dengan rasio perbandingan 1.76 : 0.61 detik pada jumlah kata uji sebanyak 4.

Kata Kunci: Mesin Pencarian, Perbaikan Kata, *Jaro-Winkler Distance*, *Multiprocessing*.

JARO-WINKLER DISTANCE FOR SPELLING CORRECTION USING MULTIPROCESSING IN TOURISM NEWS SEARCH ENGINE SYSTEMS

ABSTRACT

The development of information technology is currently changing people's lifestyles, one of which is in terms of seeking information, including tourism news as an example. In deciding to go on vacation somewhere, tourists need some good news references in the form of rubrics or online articles. Tourists can search for these references through tourism news portals using a search engine then the system will provide feedback based on the keywords provided. However, in reality, often the results given by the system do not match the interpretation intended by the user, several contributing factors are errors in providing keywords that do not include standard spelling or typos. This can be handled by building a system that is able to correct typing errors so that they comply with writing conventions using the Jaro-Winkler Distance algorithm which can be used to calculate the similarity distance between two strings, then evaluate it by applying multiprocessing to increase system execution time. The test results show that the Jaro-Winkler Distance algorithm is capable of correcting and providing inaccurate word suggestions with an average accuracy of all types of errors of 0.86. Furthermore, testing of multiprocessing shows that the parallel processing model is able to speed up execution time on each device tested. The shortest execution time is found on MacBook Pro devices both in serial and parallel processes with a ratio of 1.76:0.61 seconds for the number of test words of 4.

Keyword: Search Engine, Word Repair, Jaro-Winkler Distance, Multiprocessing.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR	iv
ABSTRAK	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
DAFTAR PERSAMAAN	xiii
DAFTAR KODE PROGRAM.....	xiv
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.2.1 Permasalahan.....	3
1.2.2 Metode Usulan	4
1.2.3 Pertanyaan Penelitian	4
1.3 Tujuan dan Manfaat	4
1.3.1 Tujuan Penelitian.....	4
1.3.2 Manfaat.....	5
1.4 Batasan Masalah.....	5
1.5 Sistematika Penulisan.....	6
BAB 2 TINJAUAN PUSTAKA	8
2.1 Kesalahan Penulisan.....	8
2.2 <i>Search engine</i>	8
2.3 <i>Text Mining</i>	8
2.4 <i>Approximate String Matching</i>	10
2.5 <i>Algoritma Jaro-Winkler Distance</i>	10
2.6 <i>Text Preprocessing</i>	12
2.7 <i>Parallel processing</i>	12

2.8	<i>Precision and Recall</i>	13
2.9	Penelitian Terkait	14
BAB 3 METODE USULAN.....		18
3.1	<i>Dataset</i>	18
3.2	Arsitektur Sistem.....	18
3.2.1	Diagram Alur <i>Preprocessing</i>	19
3.2.2	Diagram <i>Jaro-Winkler Distance</i>	21
3.2.3	Diagram <i>Multiprocessing</i>	23
3.3	Perhitungan Metode	24
3.4	Perhitungan Akurasi	25
3.5	Skenario Pengujian.....	25
3.6	Tahapan Penelitian	25
3.6.1	Identifikasi Masalah	26
3.6.2	Studi Literatur	26
3.6.3	Perancangan Sistem.....	27
3.6.4	Implementasi	27
3.6.5	Pengujian dan Analisis	27
3.6.6	Kesimpulan dan Saran.....	27
BAB 4 HASIL DAN PEMBAHASAN.....		29
4.1	Lingkungan Pengujian.....	29
4.2	Implementasi <i>Dataset</i>	31
4.3	Implementasi Sistem	31
4.3.1	<i>Preprocessing</i>	32
4.3.2	<i>Jaro-Winkler Distance</i>	34
4.3.3	<i>Multiprocessing</i>	37
4.4	Pengujian dan Analisis	41
4.4.1	Hasil Skenario Pengujian Pertama	42
4.4.2	Hasil Skenario Pengujian Kedua.....	48
4.5	Tampilan Antarmuka.....	50
4.5.1	Tampilan Awal Aplikasi	50
4.5.2	Tampilan Hasil Pencarian	50
4.6	Evaluasi Sistem	52

BAB 5 PENUTUP	55
5.1 Kesimpulan.....	55
5.2 Saran.....	55
DAFTAR PUSTAKA	57
LAMPIRAN.....	60
DAFTAR RIWAYAT HIDUP.....	80

DAFTAR GAMBAR

Gambar 3.1 Diagram Arsitektur Sistem.....	19
Gambar 3.2 Diagram Alur <i>Preprocessing</i>	20
Gambar 3.3 Diagram Tahapan Algoritma <i>Jaro-Winkler Distance</i>	22
Gambar 3.4 Diagram Sistem <i>Multiprocessing</i> [16]	23
Gambar 3.5 Diagram Alur Tahapan Penelitian.....	26
Gambar 4.1 Perangkat Keras Terpakai	29
Gambar 4.2 Tampilan <i>Dashboard</i>	50
Gambar 4.3 Tampilan Hasil Pencarian Dengan <i>Keyword</i> Salah Ketik.....	51
Gambar 4.4 Tampilan Hasil Pencarian Dengan <i>Keyword</i> Benar	51
Gambar 4.5 Grafik Hasil Pengujian Pertama	52
Gambar 4.6 Grafik Hasil Pengujian Kedua.....	53

DAFTAR TABEL

Tabel 1.1 Daftar Spesifikasi <i>Hardware</i> yang Digunakan	5
Tabel 2.1 Tabel Kontingensi	14
Tabel 2.2 Penelitian Terkait	16
Tabel 3.1 Contoh Tahapan <i>Preprocessing</i>	21
Tabel 4.1 Perangkat Lunak Terpakai	30
Tabel 4.2 Pengujian Kesalahan Penulisan Penambahan Huruf	42
Tabel 4.3 Pengujian Kesalahan Penulisan Penghapusan Huruf	44
Tabel 4.4 Pengujian Kesalahan Penulisan Penggantian Huruf	45
Tabel 4.5 Pengujian Kesalahan Penulisan Penukaran Huruf	47
Tabel 4.6 Hasil Pengujian Pertama	48
Tabel 4.7 Kata Uji Pada Pengujian Kedua	49
Tabel 4.8 Hasil Pengujian Kedua	49

DAFTAR PERSAMAAN

2.1 Persamaan <i>Jaro Distance</i>	11
2.2 Persamaan Jarak Teoritis Karakter yang Sama Dapat Dibenarkan.....	11
2.3 Persamaan <i>Jaro-Winkler Distance</i>	11
2.4 Persamaan <i>Precision</i>	13
2.5 Persamaan <i>Recall</i>	14
3.1 Persamaan <i>Accuracy</i>	25

DAFTAR KODE PROGRAM

Kode Program 4.1 <i>Preprocessing Text</i>	32
Kode Program 4.2 <i>Algoritma Jaro-Winkler Distance</i>	34
Kode Program 4.3 <i>Multiprocessing</i>	37

DAFTAR LAMPIRAN

Lampiran 1. <i>Dataset</i> Berita Pariwisata	60
Lampiran 2. <i>User Relevance Judgement</i>	72

BAB 1

PENDAHULUAN

Bab ini akan menjelaskan mengenai latar belakang, perumusan masalah, manfaat dan tujuan, batasan masalah, dan sistematika penulisan tugas akhir yang menjadi dasar pembuatan sistem koreksi ejaan kata bahasa Indonesia yang menerapkan *multiprocessing* yang merupakan salah satu pendekatan *parallel processing* pada sistem pencarian berita pariwisata menggunakan algoritma *Jaro-Winkler Distance*.

1.1 Latar Belakang

Era perkembangan teknologi informasi menuntut perubahan fundamental dalam ranah industri dan perilaku manusia yang telah memasuki era baru. Era yang menjurus ke *machine learning*, komputasi kognitif, *Internet of Things* (IoT), dan pertukaran data secara cepat dan mudah. Dampak dari perubahan ini juga selaras dengan manusia selaku peran penggerak di dalamnya. Semua aspek media informasi tak luput serentak untuk beranjak menjajal determinasi penggunaan internet ini [1]. Bermula untuk mencari informasi yang sering didapatkan dari radio, televisi, koran, ataupun media lainnya kini semuanya tersedia di internet, termasuk berita pariwisata sebagai contohnya.

Seorang wisatawan dalam memutuskan untuk liburan ke suatu tempat memerlukan beberapa referensi berita baik berupa rubrik atau artikel online. Salah satu cara yang dapat dilakukan wisatawan untuk mencari referensi tersebut adalah melalui portal berita pariwisata dengan menggunakan mesin pencarian. Hanya dengan menginputkan kata kunci (*keyword*) pada mesin pencarian (*search engine*) kemudian sistem akan memberikan umpan balik berdasarkan kata kunci yang diberikan. Namun seringkali hasil yang didapatkan wisatawan tersebut tidak sesuai dengan apa yang diharapkan, beberapa penyebabnya adalah kesalahan dalam memberikan *keyword* yang tidak termasuk ejaan baku atau terdapat kesalahan ketik (*typo*) [2].

Terdapat kaidah penulisan resmi yang harus diperhatikan contohnya pemakaian kata baku dan tidak baku. Kesalahan pengetikan yang terjadi dapat

mengakibatkan kata baku berubah menjadi kata tidak baku disebabkan ejaan yang tidak sesuai dengan Ejaan Yang Dibenarkan (EYD) yang telah ditetapkan. Sebagai contoh *user* memasukkan kata kunci “projek” padahal ejaan bakunya adalah “proyek”. Beberapa penyebab yang menjadi faktor terjadinya kesalahan pengetikan ini adalah letak huruf pada *keyboard* yang berdekatan, *slip* pada jari tangan, kesalahan lantaran ketidaksengajaan, dan ketidaktahuan kata baku yang mengakibatkan *user* mendapatkan informasi yang tidak sesuai, kurang lengkap, atau bahkan gagal.

Pendekatan yang biasa diterapkan untuk mengidentifikasi kesalahan pengetikan yaitu metode pencocokan *string* berdasarkan jarak terdekat. Pencocokan *string* (*string matching*) didefinisikan sebagai permasalahan menemukan susunan pola pada *string* di dalam *string* lain atau bagian dari isi teks [3]. Pencocokan *string* adalah bagian penting dari proses pencarian *string* (*string searching*) dalam suatu dokumen. Hasil pencarian *string* dalam sebuah dokumen bergantung pada metode pencocokan *string* yang diterapkan [4].

Pencocokan *string* umumnya dikelompokkan menjadi dua jenis, yaitu *exact string matching* (pencocokan *string* secara tepat dengan tatanan karakter dalam *string* yang dicocokkan) dan *inexact string matching* (pencocokan *string* secara samar, yakni pencocokan *string* di mana *string* yang dibandingkan memiliki kesamaan namun berbeda dalam hal susunan karakter) [5]. Dua pendekatan pada *inexact string matching* mencakup *approximate string matching* yang mencocokkan *string* berdasarkan kemiripan penulisan dan *phonetic string matching* yang mencocokkan *string* dalam hal kesamaan pengucapan. *Approximate string matching* dapat digunakan untuk pencarian *string* berdasarkan *string* yang memiliki kesamaan dan *string* yang memiliki kemiripan penulisan dengan yang ada pada kamus [6]. Metode ini bisa diterapkan pada pencarian kata baku karena dapat mengenali *string* yang identik dan yang memiliki ejaan penulisan yang mirip.

Algoritma yang bisa diterapkan untuk pencarian *string* dalam *approximate string matching* adalah *Jaro-Winkler Distance*, *Levenshtein Distance*, *Damerau Levenshtein Distance* (pengembangan dari *Levenshtein Distance*), dan *Hamming*

Distance. Masing-masing algoritma ini memiliki rumus dan perhitungan yang digunakan untuk memproses pencarian *string* dengan tingkat akurasi yang berbeda.

Sebelumnya telah dilakukan penelitian yang membandingkan 4 algoritma dalam metode *Approximate String Matching* yaitu algoritma *Jaro-Winkler Distance*, *Levenshtein Distance*, *Damerau Levenshtein Distance*, dan *Hamming Distance*. Penelitian tersebut mengatakan jika algoritma *Jaro-Winkler Distance* memiliki nilai tertinggi dibandingkan dengan metode lainnya [7]. Selanjutnya terdapat penelitian yang menerapkan metode *Approximate String Matching* pada sistem mesin pencarian judul skripsi menggunakan algoritma *Levenshtein Distance*. Hasil penelitian tersebut menyebutkan algoritma *Levenshtein* dapat menangani permasalahan pada kesalahan ketik *keyword* dengan mekanisme jenis kesalahan penambahan, penghapusan, dan penyisipan karakter huruf [8].

Pada penelitian mengoreksi ejaan kata sistem mesin pencarian berita pariwisata ini menggunakan beberapa parameter yaitu masing-masing panjang dari kedua *string* yang akan dibandingkan, jumlah *string* yang sama, ada tidaknya transposisi karakter, panjang prefiks di awal *string* sebelum ditemukan ketidaksamaan, dan yang terakhir nilai konstanta *scaling factor*. Penelitian ini menggunakan 61.434 daftar kata bahasa Indonesia yang diambil dari Kamus Besar Bahasa Indonesia (KBBI) dan 30.000 daftar kata bahasa Inggris paling umum digunakan yang telah dikompilasi oleh Peter Norvig, kamus bahasa Inggris ini diperlukan karena banyak istilah dalam pariwisata yang menggunakan bahasa Inggris. Algoritma yang diterapkan adalah *Jaro-Winkler Distance* dengan harapan dihasilkan suatu sistem yang dapat melakukan koreksi ejaan dan memiliki tingkat akurasi yang baik pada empat jenis kesalahan penulisan, yaitu kesalahan yang disebabkan penambahan, penghapusan, penggantian, dan penukaran huruf. Perbedaan penelitian ini dan penelitian sebelumnya adalah implementasi *multiprocessing* untuk meningkatkan waktu eksekusi sistem.

1.2 Rumusan Masalah

1.2.1 Permasalahan

Kata kunci (*keyword*) yang diberikan pada mesin pencarian (*search engine*) dan hasil umpan balik oleh sistem tidak sesuai dengan interpretasi yang dimaksud

oleh *user*. Hal ini bisa terjadi karena kesalahan dalam memasukkan kata kunci yang bukan kata baku atau terjadi kesalahan ketik. Kemudian waktu eksekusi yang cenderung lama yang disebabkan *dataset* yang cukup besar sehingga perlu dilakukan efisiensi sistem dengan mempercepat waktu eksekusi.

1.2.2 Metode Usulan

Berdasarkan pemaparan latar belakang dan permasalahan di atas, maka metode usulan pada penelitian ini adalah *Approximate String Matching* dengan algoritma *Jaro-Winkler Distance* dan evaluasi waktu eksekusi sistem menggunakan *multiprocessing*.

1.2.3 Pertanyaan Penelitian

Dari uraian yang telah dipaparkan di atas, didapatkan pertanyaan penelitian sebagai berikut:

1. Berapa akurasi identifikasi kesalahan pengetikan teks bahasa Indonesia dan saran perbaikan kata yang dihasilkan menggunakan algoritma *Jaro-Winkler Distance*?
2. Berapa rasio perbandingan waktu eksekusi yang dibutuhkan pada pemrosesan sistem sebelum dan sesudah menerapkan *multiprocessing* di beberapa *hardware* dengan spesifikasi berbeda?

1.3 Tujuan dan Manfaat

1.3.1 Tujuan Penelitian

Tujuan perancangan sistem identifikasi salah ketik dan perbaikan kata pada mesin pencarian berita pariwisata adalah sebagai berikut:

1. Mengimplementasikan algoritma *Jaro-Winkler Distance* untuk identifikasi dan perbaikan kata teks bahasa Indonesia.
2. Mengetahui akurasi yang dihasilkan algoritma *Jaro-Winkler Distance* dalam menentukan relevansi saran kata sesuai kamus.
3. Menerapkan *multiprocessing* untuk meningkatkan efisiensi waktu eksekusi sistem.
4. Untuk mengetahui nilai rasio perbandingan antara waktu eksekusi pemrograman secara serial dan secara paralel.

1.3.2 Manfaat

Adapun manfaat diharapkan pada perancangan sistem ini adalah hasil sistem yang dibuat mampu memberikan umpan balik sesuai dari kata kunci yang diberikan dengan interpretasi yang dimaksud oleh pengguna.

1.4 Batasan Masalah

Ruang lingkup atau batasan ini diperlukan agar dalam pengerjaannya tidak melebar dari sasaran yang diteliti. Adapun batasan masalah dalam pembuatan sistem ini adalah sebagai berikut:

1. Metode identifikasi salah ketik dan perbaikan kata menggunakan *Jaro-Winkler Distance*.
2. Pengujian dilakukan dengan membagi ke dalam empat jenis kesalahan penulisan, yaitu kesalahan yang disebabkan penghapusan, penggantian, penambahan, dan penukaran huruf.
3. Sistem ini hanya akan mengoreksi kata yang terdapat pada kamus bahasa Indonesia [9] dan kata bahasa Inggris paling umum digunakan yang telah dikompilasi oleh Peter Norvig [10].
4. *Output* berupa penemuan kesalahan penulisan dalam data input pada mesin pencarian dan saran perbaikan kata yang telah disesuaikan pada kamus.
5. Implementasi sistem yang dibangun adalah mesin pencarian berita pariwisata.
6. Evaluasi sistem untuk mempercepat waktu eksekusi sistem menggunakan *multiprocessing*.
7. Spesifikasi perangkat yang digunakan dapat dilihat pada Tabel 1.1.

Tabel 1.1 Daftar Spesifikasi *Hardware* yang Digunakan

<i>Hardware</i>	<i>CPU</i>	<i>Core & Threads</i>	<i>RAM</i>	<i>Operation System</i>
Asus VivoBook	AMD Ryzen 7 3700U	4 <i>cores</i> & 8 <i>threads</i>	8 GB	Windows
HP G4 MT Business PC	Intel i7-8700	6 <i>cores</i> & 12 <i>threads</i>	8 GB	Linux Ubuntu
Macbook Pro	Apple M1	8 <i>cores</i> & 8 <i>threads</i>	16 GB	MacOS

1.5 Sistematika Penulisan

Sistematika yang diterapkan dalam penyusunan laporan tugas akhir ini digunakan untuk memberikan gambaran umum mengenai apa saja yang akan dibahas pada penelitian ini, sehingga secara sekilas pembaca akan mengetahui garis besar dan poin-poin penting dalam penelitian ini. Sistematika penulisan terbagi dalam beberapa pokok bahasan sebagai berikut:

BAB I PENDAHULUAN

Bab ini menguraikan latar belakang, rumusan masalah, tujuan dan manfaat dari penelitian, batasan masalah, dan sistematika penulisan laporan penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan teori yang sudah ada atau dari penelitian-penelitian yang telah dilakukan dengan tujuan untuk mendukung serta mendasari sebuah sistem yang akan dibangun. Dasar teori sangat dibutuhkan demi tercapainya suatu tujuan sistem yang akan dibuat dan belajar dari kekurangan penelitian sebelumnya. Dasar teori yang akan digunakan pada penelitian ini adalah kesalahan penulisan, *search engine*, *text mining*, *Approximate String Matching*, algoritma *Jaro-Winkler Distance*, *text preprocessing*, *parallel processing*, *precision and recall*, dan penelitian terkait.

BAB III METODE PENELITIAN

Bab ini menyajikan garis besar mengenai metode yang digunakan dalam penelitian. Dimulai dari *dataset*, arsitektur sistem, skenario pengujian, perhitungan metode, perhitungan akurasi, dan tahapan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjabarkan hasil dari uji coba pengimplementasian sistem berdasarkan perancangan dan metodologi yang telah

dijelaskan pada bab sebelumnya. Terdapat juga ulasan mengenai tahapan pembangunan sistem dimulai dari persiapan *tools* yang digunakan hingga pengujian sistem beserta analisis dan evaluasi yang dihasilkan.

BAB V

PENUTUP

Bab ini berisikan kesimpulan dari pembuatan sistem koreksi kata pada sistem pencarian berita pariwisata dan juga saran yang diharapkan dapat menjadi masukan untuk menjadikan sistem ini lebih baik lagi untuk kedepannya.

BAB 2

TINJAUAN PUSTAKA

2.1 Kesalahan Penulisan

Kesalahan penulisan adalah kesalahan yang kerap kali dilakukan manusia saat melakukan kegiatan yang berkaitan dengan hal menulis. Kesalahan penulisan itu sendiri terbagi dalam dua kategori yaitu kesalahan yang disadari dan tidak disadari. Kesalahan penulisan yang dilakukan dengan sengaja dapat terjadi karena hal tersebut bersifat berulang dan terjadi secara terus menerus, sebagai contoh kata "dalam disingkat dlm", "jalan disingkat jln" dan kata lain yang sering ditulis tidak beraturan atau tidak baku sesuai kaidah penulisan resmi yang telah disepakati. Sedangkan untuk kesalahan penulisan yang tidak disengaja atau sering disebut dengan istilah tipografi (*typographical error*) [11]. Kesalahan penulisan yang disebabkan tipografi ini meliputi pergeseran urutan karakter serta perubahan karakter dalam satu kata yang biasa terjadi karena kesalahan pengetikan pada *keyboard*. Sehingga dapat dikelompokkan menjadi 4 jenis kesalahan penulisan yaitu: penambahan, penghapusan, penggantian, dan penukaran huruf [12].

2.2 Search engine

Mesin pencarian atau biasa disebut dengan *search engine* adalah program yang digunakan sebagai alat untuk mencari informasi di internet. Mesin pencari memiliki *database* elektronik dari jutaan hingga miliaran alamat situs web dan informasi yang terakumulasi di internet [13]. Cara menggunakan mesin pencari adalah dengan memasukkan kata kunci (*keyword*) yang ingin dicari, kemudian akan ditampilkan beberapa tautan yang mengarah ke situs atau informasi yang saling terhubung dengan kata kunci yang dimasukkan. Fungsi mesin pencari tidak lain untuk mencari informasi dengan memasukkan kata kunci yang dimaksud, sehingga dengan mudah dan cepat menemukan situs web yang berisi berbagai bentuk informasi seperti teks, video, foto, dan lain-lain.

2.3 Text Mining

Pemrosesan teks atau dalam istilah umum disebut dengan istilah penambangan teks (*Text Mining*) adalah salah satu dari banyak bidang kecerdasan

buatan di mana ia menerapkan konsep yang sama pada *data mining* untuk mencari pola teks yang melalui proses ekstraksi terlebih dahulu agar dapat diambil informasi dari pola data teks yang semula tidak terstruktur.

Teknik yang digunakan *text mining* adalah mengumpulkan data berupa teks dari berbagai sumber seperti dokumen, dengan tujuan untuk menemukan esensi dari dokumen tersebut guna dapat dilakukan proses analisis relasi dan kemiripan dengan dokumen lainnya [14]. Terdapat 7 jenis *Text Mining* yang ada saat ini, sebagai berikut [15]:

1. *Search and Information Retrieval (IR)*

Sistem penyimpanan dan temu kembali untuk dokumen teks, termasuk pencarian kata kunci pada mesin pencarian, dan penambahan teks seperti deteksi kesalahan ketik.

2. *Information Extraction (IE)*

Ekstraksi dan identifikasi dari fakta-fakta yang relevan dan hubungan dari teks yang tidak terstruktur; proses pembentukan data terstruktur dari teks yang semi-terstruktur dan tidak terstruktur.

3. *Document Classification*

Pengategorian dan pengelompokan potongan kata, istilah, paragraf, atau dokumen berdasarkan model yang telah dilatih pada data berlabel yang menggunakan metode klasifikasi *data mining*.

4. *Document Clustering*

Pengategorian dan pengelompokan potongan kata, istilah, paragraf, atau dokumen berdasarkan model yang telah dilatih pada data yang tidak berlabel yang menggunakan metode klasifikasi *data mining*.

5. *Web Mining*

Text Mining data pada internet yang menitik beratkan skala dan keterhubungan antar web.

6. *Natural Language Processing*

Pemrosesan bahasa atau linguistik komputasional yang mengkaji bahasa alami manusia pada perangkat komputasi seperti memahami, menafsirkan, serta memanipulasi perintah manusia.

7. *Concept Extraction*

Pengategorian frasa dan kata ke dalam kelompok yang secara semantik serupa.

2.4 *Approximate String Matching*

Metode *Approximate String Matching* adalah teknik pencarian *string* berdasarkan jarak kesamaan dari sudut penulisan (banyaknya karakter, susunan karakter dalam dokumen). Tingkat kemiripan tergantung pada kedekatan penulisan dari kedua *string* yang dibandingkan [6]. Metode ini merupakan turunan dari *inexact string matching* yang memiliki dua pendekatan, yakni *approximate string matching* yang mencocokkan *string* berdasarkan kemiripan penulisan dan *phonetic string matching* yang mencocokkan *string* dalam hal kesamaan pengucapan [6].

Algoritma di dalam *approximate string matching* yang dapat diterapkan untuk pencarian *string* diantaranya adalah *Jaro-Winkler Distance*, *Levenshtein Distance*, *Damerau Levenshtein Distance* (pengembangan dari *Levenshtein Distance*), dan *Hamming Distance*. Penerapan dari metode ini bisa dimanfaatkan pada pencarian kata baku karena dapat mengenali *string* yang identik dan yang memiliki ejaan penulisan yang mirip. Pada penelitian ini algoritma yang digunakan untuk mencari dan menghitung kedekatan *string* adalah *Jaro-Winkler Distance*.

2.5 *Algoritma Jaro-Winkler Distance*

Algoritma *Jaro-Winkler Distance* adalah algoritma untuk mengukur kemiripan dari dua irisan *string* dan merupakan varian *Jaro Distance Metric* [2]. Algoritma ini juga dapat diterapkan untuk deteksi duplikat. Semakin tinggi nilai *Jaro-Winkler Distance* pada kedua *string*, maka semakin mirip *string* tersebut. Nilai 0 menunjukkan tidak ada kecocokan antar *string*, dan nilai 1 menandakan sama persis [16].

Algoritma ini memiliki tiga bagian dasar, yaitu [17]:

1. Menghitung panjang dari *string*
2. Menemukan jumlah karakter yang sama di antara kedua *string*
3. Menemukan jumlah transposisi

Algoritma *Jaro-Winkler Distance* menggunakan rumus untuk menghitung jarak (dj) antara kedua *string* yaitu s_1 dan s_2 , Sebagaimana tertera pada persamaan 2.1 [16]:

$$dj = \frac{1}{3} \times \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m - t}{m} \right) \quad 2.1$$

Keterangan:

m = jumlah karakter yang sama di kedua *string*

$|s_1|$ = panjang *string* 1

$|s_2|$ = panjang *string* 2

t = jumlah transposisi

Dua buah karakter memiliki jarak teoritis yang disamakan dapat dibenarkan apabila tidak melebihi pada persamaan 2.2 [16]:

$$\left(\frac{\max(|s_1|, |s_2|)}{2} \right) - 1 \quad 2.2$$

Dengan adanya jarak teoritis ini, meskipun terdapat karakter yang sama pada kedua kata yang dibandingkan tetapi letak karakter tersebut di luar jangkauan jarak teoritis maka tetap tidak dihitung sebagai karakter yang sama [18].

Algoritma *Jaro-Winkler Distance* menggunakan *prefix scale* (p), dengan konstanta nilai standar menurut Winkler adalah $p = 0,1$. Untuk *prefix length* (l), yaitu merepresentasikan panjang karakter yang sama sampai ditemukan adanya ketidakmiripan (dengan maksimal 4 karakter), sehingga ditemukan perhitungan nilai *Jaro-Winkler Distance* (dw), sebagaimana tercantum pada persamaan 2.3 [16]:

$$dw = dj + (lp(1 - dj)) \quad 2.3$$

Keterangan:

dj = *jaro distance* untuk *string* s_1 dan s_2

l = panjang prefiks umum di awal *string* nilai maksimal 4 karakter
(panjang karakter yang sama sebelum ditemukan ketidaksamaan)

p = konstanta *scaling factor*. Nilai standar menurut Winkler = 0,1

2.6 *Text Preprocessing*

Dokumen pada umumnya mempunyai struktur yang tidak beraturan atau tanpa struktur. Oleh karena itu, diperlukan suatu proses yang dapat mengubah bentuk data dari sebelumnya tidak terstruktur menjadi bentuk data yang terstruktur. Proses pengubahan ini umumnya dikenal dengan istilah *preprocessing* [19].

Tujuan dilakukannya proses *preprocessing* ini agar data dapat lebih mudah diproses oleh sistem yang kemudian akan menghasilkan data *term* (data yang sudah melalui proses *preprocessing*).

Preprocessing data pada teks terdiri dari beberapa proses tahapan sebagai berikut [19]:

1. *Case Folding*, digunakan untuk menyamaratakan semua huruf pada teks dari huruf kapital (*upper case*) menjadi huruf kecil (*lower case*).
2. *Cleaning*, berfungsi dalam membersihkan karakter tertentu yang tidak dibutuhkan seperti simbol dan angka.
3. *Tokenizing*, digunakan untuk menguraikan kalimat menjadi kata perkata berdasarkan karakter 'spasi' sebagai tanda pemisahannya.
4. Normalisasi kata, yaitu mengubah kata yang tidak baku menjadi kata yang baku sesuai yang tertera pada kamus.
5. *Stopword removal*, digunakan untuk menghilangkan kata yang dianggap tidak penting ketika dilakukan pengolahan data teks karena data *stopword* dianggap tidak berpengaruh terhadap kualitas hasil sistem. Hal ini dilakukan untuk mengurangi waktu pemrosesan.
6. *Stemming*, merupakan proses pengembalian bentuk kata menjadi kata dasar dengan menghilangkan imbuhan.

2.7 *Parallel processing*

Pemrograman paralel dapat didefinisikan sebagai model yang bertujuan untuk membuat program yang kompatibel dengan lingkungan yang disiapkan untuk mengeksekusi instruksi kode secara bersamaan. Dasar yang kuat dalam pemrograman paralel memungkinkan pengoptimalan kinerja aplikasi. Hasil yang diperoleh berupa peningkatan pengalaman pengguna serta memaksimalkan

ketersediaan sumber daya komputasi, sehingga sistem memerlukan lebih sedikit waktu pemrosesan untuk menyelesaikan tugas yang kompleks. Model pemrosesan paralel ini cocok pada skenario *database* dengan ukuran yang cukup besar [20].

Pada bahasa pemrograman *python* salah satu *library* yang menerapkan model *parallel processing* adalah *multiprocessing*. *Multiprocessing* mendukung sistem untuk menjalankan tugas di beberapa unit pemrosesan (*cores*) pada waktu bersamaan. Kata "multi" dalam *multiprocessing* mengacu pada beberapa *core* di unit pemrosesan pusat (CPU) komputer yang berarti bahwa memanfaatkan lebih dari satu inti dapat secara dramatis meningkatkan waktu pemrosesan. Jumlah maksimum proses yang dapat dikerjakan tergantung jumlah prosesor pada komputer. Ada dua objek utama dalam *package multiprocessing* pada *python* yaitu *pool class* dan *process class*. *Pool class* lebih efisien diterapkan pada kasus dengan jumlah proses yang besar. Sedangkan *process class* lebih efisien pada situasi yang terdapat banyak inputan namun jumlah proses yang dilakukan tidak banyak [21].

2.8 Precision and Recall

Pengukuran efektivitas temu kembali informasi (*information retrieval*) dapat dilakukan dengan menghitung nilai presisi atau ketepatan (*precision*) dan nilai perolehan (*recall*). *Precision* dapat dimaknai sebagai kecocokan antara permintaan informasi kata kunci yang diajukan. Saat seseorang mencari informasi dalam sistem dan sistem menampilkan banyak dokumen, maka pencocokan dokumen tersebut disebut relevansi. Sedangkan *Recall* adalah proporsi jumlah dokumen yang dapat ditemukan kembali oleh proses pencarian sistem temu kembali informasi [22].

Precision (ketepatan) adalah rasio antara dokumen yang relevan dan jumlah dokumen yang ditemukan dalam pencarian. Presisi berkaitan dengan kemampuan sistem untuk tidak memanggil dokumen yang tidak relevan. Untuk menghitung nilai presisi, digunakan persamaan sebagai berikut [23]:

$$Precision = \frac{\text{jumlah dokumen relevan yang terambil}}{\text{jumlah dokumen terambil dalam pencarian}} \quad 2.4$$

Recall (perolehan) adalah rasio perbandingan dokumen yang ditemukan dengan keseluruhan dokumen yang relevan yang terdapat dalam sistem. *Recall*

identik dengan kemampuan sistem untuk mengambil kembali dokumen yang relevan. Untuk menghitung nilai *recall*, dapat gunakan persamaan berikut [23]:

$$Recall = \frac{\text{jumlah dokumen relevan yang terambil}}{\text{jumlah dokumen relevan pada database}} \quad 2.5$$

Berdasarkan persamaan 2.4 mengenai *precision*, dan persamaan 2.5 mengenai *recall* maka dapat dirumuskan tabel kontingensi sebagai berikut:

Tabel 2.1 Tabel Kontingensi

		Nilai Sebenarnya	
		<i>Relevance</i>	<i>Not Relevance</i>
Nilai Prediksi	<i>Retrieved</i>	TP (<i>True Positive</i>)	FP (<i>False Positive</i>)
	<i>Not Retrieved</i>	FN (<i>False Negatif</i>)	TN (<i>True Negatif</i>)

Keterangan:

- TP = dokumen relevan dengan jumlah dokumen yang ditemu balik dalam penelusuran
- FP = dokumen tidak relevan dengan jumlah dokumen yang ditemu balik dalam penelusuran
- FN = dokumen relevan dengan jumlah dokumen yang tidak ditemu balik dalam penelusuran
- TN = dokumen tidak relevan dengan jumlah dokumen yang tidak ditemu balik dalam penelusuran

2.9 Penelitian Terkait

Berdasarkan penelitian yang ada, penulis merujuk beberapa penelitian sebelumnya yang cakupan ranahnya saling berdekatan atau identik guna mendukung dan membantu pengembangan penelitian yang dilakukan lebih lanjut.

Yeny Rochmawati, dkk [7] melakukan penelitian untuk membandingkan 4 algoritma pada metode *Approximate String Matching* yaitu algoritma *Jaro-Winkler Distance*, *Levenshtein Distance*, *Damerau Levenshtein Distance*, dan *Hamming*

Distance. Hasil dari penelitian ini adalah algoritma *Jaro-Winkler Distance* memiliki nilai paling tinggi dibandingkan algoritma lainnya, dengan nilai MAP sebesar 0,87 terbagi ke dalam empat jenis kesalahan penulisan yaitu penukaran huruf 0,95, kesalahan penambahan huruf 0,90, kesalahan penghapusan huruf 0,92, dan yang terakhir kesalahan penggantian huruf 0,70.

Anna Kurniawati, dkk [16] melakukan penelitian untuk menghitung kemiripan antar dokumen dengan cepat, untuk memprediksi plagiarisme dalam penulisan dokumen. Hasil yang diperoleh dalam uji coba dapat berjalan dengan baik untuk memeriksa kesamaan dokumen yang identik atau sama seratus persen dari dokumen yang sama. Ini terjadi karena urutan kata-kata yang dibandingkan sangat tepat. Namun, ketika dokumen diperiksa untuk kesamaan dalam urutan yang berbeda, aplikasi tidak dapat mendeteksi kesamaan.

Ida Bagus Ketut Surya Arnawa [8] melakukan penelitian untuk mengidentifikasi kesalahan kata yang diimplementasikan pada mesin pencarian menggunakan algoritma *Levenshtein* untuk memberikan saran perbaikan kata berdasarkan jarak terdekat. Hasil dari penelitian ini menyimpulkan algoritma *Levenshtein* dapat membantu mengatasi kesalahan penulisan kata kunci dengan mekanisme penghapusan, penambahan, dan penyisipan karakter.

Agung Prasetyo, dkk [24] melakukan penelitian penerapan algoritma *Jaro-Winkler Distance* pada fitur koreksi otomatis (*autocorrect*) dan saran ejaan (*spelling suggestion*) untuk tulisan bahasa Indonesia di BMS TV. Dari hasil penelitian, sistem mampu secara otomatis mengoreksi 10 kata dengan benar dan memunculkan saran ejaan kata dengan tepat pada 49 kata dari 60 sampel kata dalam berbagai skenario kesalahan penulisan kata.

Arina Indana Fahma, dkk [11] melakukan penelitian menggunakan metode *N-gram* dan *Levenshtein Distance* untuk mengidentifikasi kesalahan ketik (*typographical error*) pada dokumen berbahasa Indonesia. Penelitian ini berhasil memperoleh nilai *precision* terbaik pada uji coba kesalahan ketik sebesar 0.97 untuk jenis penyisipan dan untuk nilai *recall* terbaik pada hasil pengujian kesalahan ketik pada jenis *substitution* sebesar.

Dalam hal ini dari penelitian terkait memiliki beberapa kekurangan dalam waktu eksekusi pencarian yang cukup lama. Perbedaan pada penelitian ini dengan penelitian sebelumnya adalah ditambahkan proses untuk mempersingkat waktu eksekusi. *Dataset* dengan jumlah yang cukup banyak yakni 61.434 kata yang diambil dari Kamus Besar Bahasa Indonesia (KBBI) dan 30.000 daftar kata bahasa Inggris paling umum digunakan yang telah dikompilasi oleh Peter Norvig [10] untuk mengantisipasi data input berbahasa Inggris. Artinya, sistem melakukan pengecekan setiap iterasi kata inputannya dengan kata yang ada di *dataset*. Dengan banyaknya jumlah perulangan dan iterasi ini, maka diperlukan performa dan kecepatan waktu eksekusi yang baik dengan menerapkan *multiprocessing* yang merupakan salah satu pendekatan dari model *parallel processing*. Rangkuman dari penelitian terkait dapat dilihat pada Tabel 2.2.

Tabel 2.2 Penelitian Terkait

Peneliti, Tahun	Permasalahan	Metode	Hasil
Yeny Rochmawati, Retno Kusumaningrum, 2015 [7]	Identifikasi kesalahan pengetikan teks bahasa Indonesia	<i>Hamming Distance, Levenshtein Distance, Damerau Levenshtein Distance, Jaro-Winkler Distance.</i>	<i>Jaro-Winkler Distance</i> memiliki nilai tertinggi dibandingkan algoritma lainnya, dengan nilai MAP sebesar 0,87 terbagi ke dalam 4 jenis kesalahan penulisan yaitu penukaran, penambahan, penghapusan, dan penggantian huruf.
Anna Kurniawati, Sulistyono, Puspitodjati, dan Sazali Rahman, 2016 [16]	Menghitung tingkat kesamaan antar dokumen bahasa Indonesia.	<i>Jaro-Winkler Distance.</i>	Aplikasi Berhasil memeriksa kemiripan dokumen yang memiliki kemiripan dan urutan kata yang sama 100%.
Ida Bagus Ketut Surya Arnawa, 2017 [8]	Koreksi ejaan kata bahasa	<i>Levenshtein Distance.</i>	Algoritma <i>Levenshtein Distance</i> dapat mengatasi permasalahan

Peneliti, Tahun	Permasalahan	Metode	Hasil
	Indonesia pada mesin pencarian.		pada kesalahan ejaan <i>keyword</i> pada mesin pencarian skripsi dengan mekanisme penambahan, penyisipan dan penghapusan karakter.
Agung Prasetyo, Wiga Maulana Baihaqi, dan Iqbaluddin Syam Had, 2018 [24]	Fitur <i>autocorrect</i> dan <i>spelling suggestion</i> pada penulisan naskah bahasa Indonesia.	<i>Jaro-Winkler Distance</i> .	Hasil penelitian mampu memunculkan saran ejaan kata dengan tepat pada 49 kata dari 60 sampel kata dari berbagai skenario kesalahan penulisan kata.
Arina Indana Fahma, Imam Cholissodin, dan Rizal Setya Perdana, 2018 [11]	Deteksi kesalahan penulisan kata dokumen berbahasa Indonesia.	<i>Levenshtein Distance, N-gram</i> .	Penelitian berhasil mendapatkan nilai presisi terbaik pada uji coba <i>typographical error</i> pada jenis <i>insertion</i> sebesar 0.97 dan untuk nilai <i>recall</i> terbaik pada jenis <i>substitution</i> sebesar 1.

BAB 3

METODE USULAN

Metode penelitian adalah langkah-langkah atau pedoman yang diterapkan penulis ketika melaksanakan penelitian, bertujuan untuk mengatur proses penelitian yang dilakukan agar terstruktur dengan sistematis.

3.1 *Dataset*

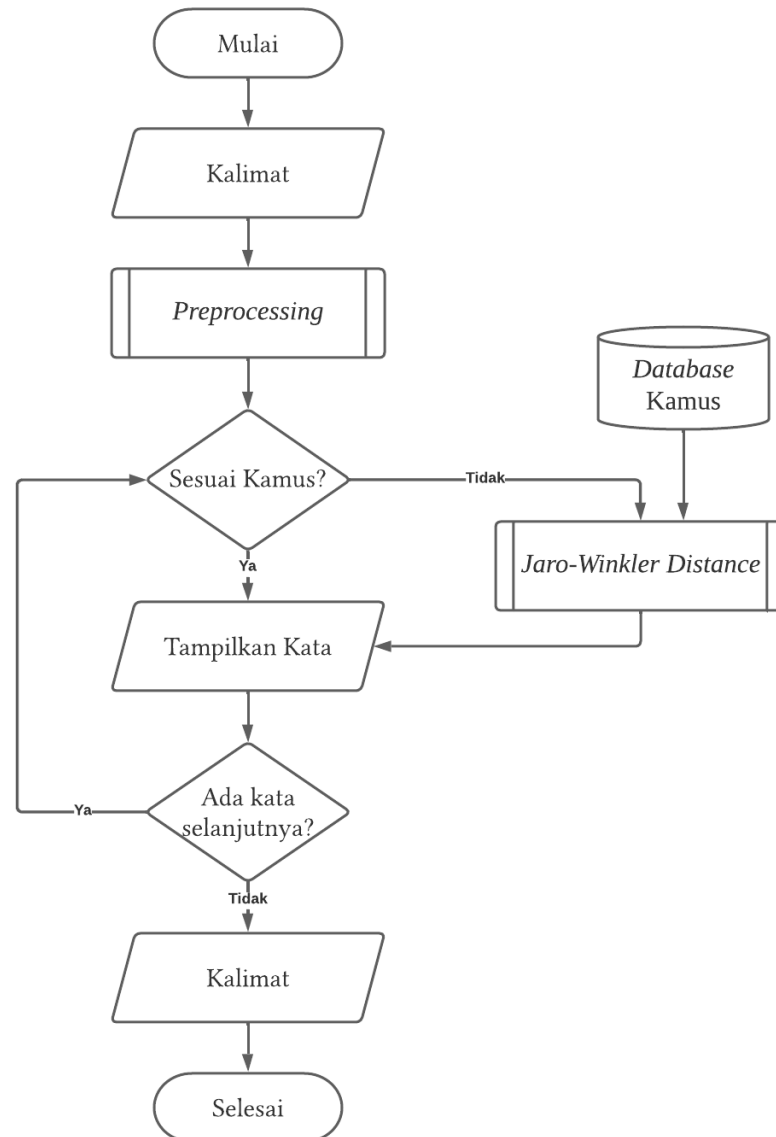
Dataset yang digunakan dalam penelitian ini berupa kosakata bahasa Indonesia yang diperoleh dari [9]. Jumlah *dataset* terdiri dari 61.434 kosakata bahasa Indonesia, 30.000 daftar kata bahasa Inggris paling umum digunakan yang telah dikompilasi oleh Peter Norvig [10], dan daftar kata *stopword* yang didapatkan dari [25], yang berjumlah 758 kata *stopword* bahasa Indonesia. Data tersebut disimpan dalam format *excel* yang kemudian di-convert ke dalam bentuk *mysql* sebagai *database*. Kemudian terdapat juga *database* berita pariwisata sebagai data yang ditampung untuk selanjutnya ditampilkan ketika *user* melakukan *request* pada sistem melalui *search engine*.

3.2 *Arsitektur Sistem*

Diagram arsitektur sistem melalui beberapa proses tahapan. Proses pertama dimulai dengan pemberian data input berupa *keyword* pada mesin pencarian oleh *user* lalu dilakukan tahapan *preprocessing* (*cleaning, case folding, tokenizing, stopwords removal*). Tujuan dilakukan proses *preprocessing* ini untuk mengekstrak data input yang semula tidak terstruktur menjadi data input yang lebih terstruktur untuk memudahkan sistem dalam melakukan pemrosesan bahasa yang kemudian akan menghasilkan daftar kata (*term list*) yang merepresentasikan setiap kata pada data input yang dimasukkan. Setelah melalui tahapan *preprocessing* sistem akan mengoreksi daftar kata sesuai dengan kamus yang ada di dalam *database*, kata yang terdapat karakter atau huruf yang kurang tepat akan dikoreksi dengan menerapkan pendekatan algoritma *Jaro-Winkler Distance* dalam mencari saran kata berdasarkan jarak *string* terdekat. Jika sistem telah mendapatkan saran kata, sistem akan mengubah kata yang kurang tepat tersebut dengan saran kata yang dihasilkan. Kemudian tahap terakhir dilakukan evaluasi sistem yang bertujuan untuk

mempersingkat efisiensi waktu eksekusi dengan menerapkan *multiprocessing* yang merupakan salah satu pendekatan dari model *parallel processing*.

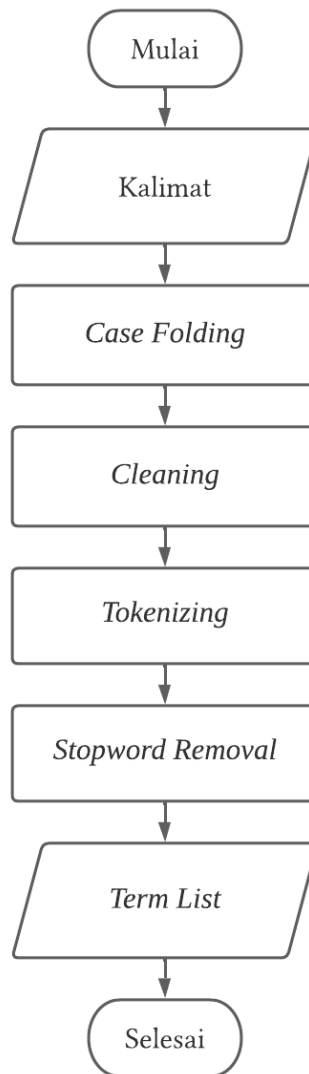
Diagram arsitektur sistem merepresentasikan garis besar alur kinerja sistem dalam koreksi salah ketik dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Arsitektur Sistem

3.2.1 Diagram Alur *Preprocessing*

Fungsi dari tahapan *preprocessing* adalah untuk mengolah data input yang semula tidak terstruktur menjadi terstruktur untuk memudahkan sistem dalam melakukan pemrosesan bahasa pada tahapan selanjutnya. Langkah-langkah tahapan sistem *preprocessing* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Alur *Preprocessing*

Seperti pada Gambar 3.2 data input yang diberikan oleh *user* kemudian akan diproses oleh sistem ke dalam tahapan *preprocessing*. Langkah pertama dalam *preprocessing* adalah *case folding*. Langkah ini dilakukan untuk menyeragamkan semua karakter menjadi huruf kecil (*lower case*), kemudian *cleaning* untuk menghilangkan simbol yang tidak diperlukan oleh sistem seperti menghilangkan tanda baca atau *whitespace* pada data input [26]. Selanjutnya *tokenizing* digunakan untuk pemecahan data input menjadi kata-kata atau *term* berdasarkan karakter spasi sebagai tanda pemisahannya. Langkah terakhir *preprocessing* yakni *stopword removal* dilakukan untuk menghapus kata yang dianggap tidak memiliki bobot atau tidak penting karena data *stopword* dinilai tidak berpengaruh terhadap kualitas hasil

sistem [27]. Contoh *preprocessing* dengan sampel data input dapat dilihat pada Tabel 3.1.

Tabel 3.1 Contoh Tahapan *Preprocessing*

Dokumen	<i>Case Folding</i>
Pantai di Sumenep.	pantai di sumenep.
D'	<i>Cleaning</i>
pantai di sumenep.	pantai di sumenep
D''	<i>Tokenizing</i>
pantai di sumenep	pantai – di – sumenep
D'''	<i>Stopword Removal</i>
pantai – di – sumenep	pantai – sumenep

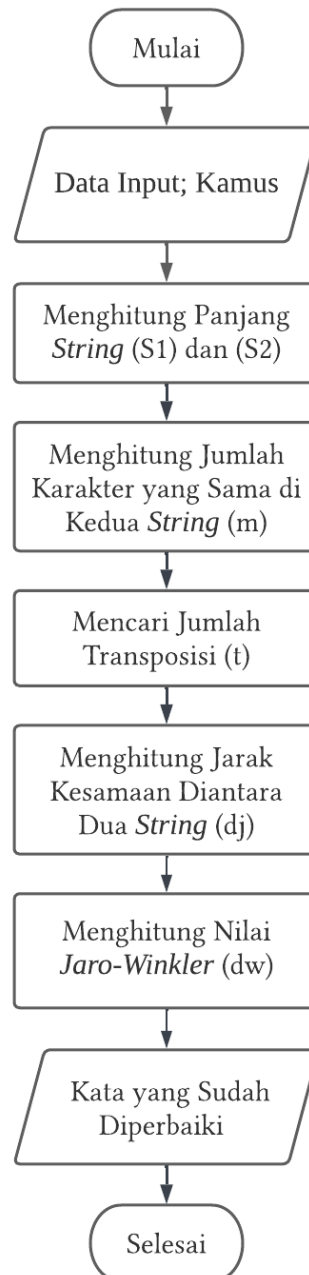
Seperti dapat dilihat pada Tabel 3.1 didapatkan *term* yaitu kata pantai dan sumenep yang selanjutnya digunakan sistem sebagai *keyword* untuk dicocokkan dengan data di *database* dan memberikan umpan balik kepada *user* sesuai data input yang diberikan.

3.2.2 Diagram *Jaro-Winkler Distance*

Perbaikan kata menggunakan algoritma *Jaro-Winkler Distance* untuk mengukur kesamaan antara dua *string*, yaitu *string* yang akan dibandingkan dan *string* acuan. Jika nilai jarak (*distance*) mendekati 1 maka *string* tersebut semakin mirip dengan *string* yang dijadikan acuan pada kamus. Begitu juga sebaliknya jika nilai jarak mendekati 0 menunjukkan ketidakmiripan dengan kata pada kamus.

Ada beberapa tahapan dalam proses ini, yaitu diawali dengan perhitungan panjang *string* dari kata input dengan kata acuan dalam KBBI. Proses kemudian dilanjutkan dengan menghitung jumlah karakter yang sama antara kedua *string* (m) berdasarkan jarak teoritis, kemudian dihitung jumlah transposisinya atau ada tidaknya karakter di dalam *string* yang sama tetapi posisinya tertukar (t). Selanjutnya hitung nilai *jaro distance* (dj) untuk menghitung jarak kesamaan di kedua *string*. Langkah terakhir hitung nilai *Jaro-Winkler Distance* (dw). Hasil *output* dari tahap ini adalah perubahan dari kata tidak baku menjadi kata baku

menurut Kamus Besar Bahasa Indonesia (KBBI). Diagram tahapan perhitungan algoritma *Jaro-Winkler Distance* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram Tahapan Algoritma *Jaro-Winkler Distance*

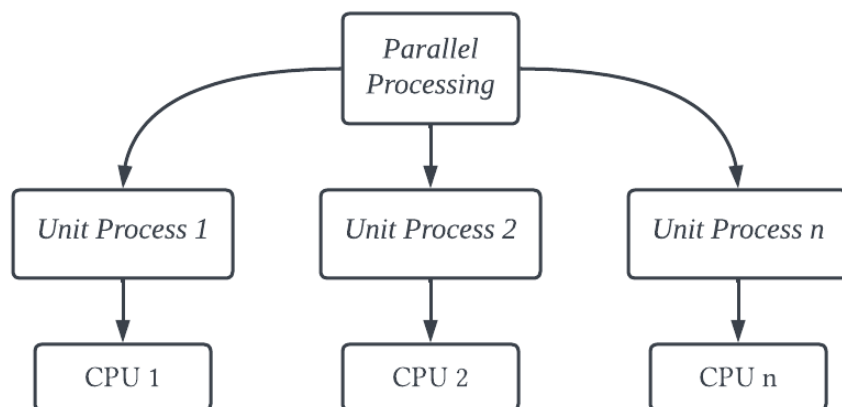
Kemudian sistem akan mengambil nilai *Jaro-Winkler distance* terbesar (1 atau yang paling mendekati) dari setiap kata pada *database* kamus yang telah dihitung jarak kedekatannya dengan string input *keyword* yang diberikan oleh *user* untuk selanjutnya dijadikan kata hasil koreksi oleh sistem.

3.2.3 Diagram *Multiprocessing*

Sistem yang dibuat membutuhkan waktu yang relatif lama disebabkan jumlah *dataset* yang cukup banyak yaitu 61.434 daftar kata yang diambil dari Kamus Besar Bahasa Indonesia (KBBI) [9] dan 30.000 daftar kata bahasa Inggris paling umum digunakan yang telah dikompilasi oleh Peter Norvig [10]. Artinya, sistem melakukan pengecekan setiap iterasi kata inputannya dengan seluruh kata yang ada di *dataset*. Dengan banyaknya jumlah perulangan dan iterasi ini perlu dilakukan evaluasi untuk meningkatkan efisiensi performa dengan mempersingkat waktu eksekusi dengan penerapan model *parallel processing* menggunakan *library* atau *package* pada bahasa pemrograman *python* salah satunya adalah *multiprocessing*.

Multiprocessing mendukung sistem untuk menjalankan tugas di beberapa unit pemrosesan (*cores*) pada waktu bersamaan. Jumlah maksimum proses yang dapat dikerjakan tergantung pada jumlah prosesor (CPU) pada perangkat komputer. Ada dua objek utama dalam *package multiprocessing* pada *python* yaitu *pool class* dan *process class*. *Pool class* lebih efisien diterapkan pada kasus dengan jumlah proses yang besar. Sedangkan *process class* lebih efisien pada situasi yang terdapat banyak inputan namun jumlah proses yang dilakukan tidak banyak [21].

Dalam hal ini, digunakan *pool class* karena sesuai dengan kebutuhan sistem yang akan dibuat pada penelitian ini. Cara kerja sistem *multiprocessing* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Diagram Sistem *Multiprocessing* [20]

3.3 Perhitungan Metode

Perhitungan manual metode dilakukan berdasarkan kata yang terdeteksi terdapat kesalahan penulisan atau tidak baku. Landasan persamaan perhitungan mengacu pada Persamaan 2.1 dan Persamaan 2.3 Berikut adalah contoh perhitungan perbaikan kata menggunakan algoritma *Jaro-Winkler Distance*:

1. Kesalahan penulisan karena penghapusan huruf

$$\text{String 1} = \text{sehinga} \quad s_1 = 7 \quad s_2 = 8 \quad m = 7$$

$$\text{String 2} = \text{sehingga} \quad t = 0 \quad l = 4 \quad p = 0.1$$

$$dj = \frac{1}{3} \times \left(\frac{7}{7} + \frac{7}{8} + \frac{7-0}{7} \right) = 0,958$$

$$dw = 0.958 + (4 \times 0.1(1 - 0.958)) = 0.975$$

2. Kesalahan penulisan karena penambahan huruf

$$\text{String 1} = \text{promossi} \quad s_1 = 8 \quad s_2 = 7 \quad m = 7$$

$$\text{String 2} = \text{promosi} \quad t = 0 \quad l = 4 \quad p = 0.1$$

$$dj = \frac{1}{3} \times \left(\frac{7}{8} + \frac{7}{7} + \frac{7-0}{7} \right) = 0.958$$

$$dw = 0.958 + (4 \times 0.1(1 - 0.958)) = 0.975$$

3. Kesalahan penulisan karena penggantian huruf

$$\text{String 1} = \text{imformasi} \quad s_1 = 9 \quad s_2 = 9 \quad m = 8$$

$$\text{String 2} = \text{informasi} \quad t = 0 \quad l = 1 \quad p = 0.1$$

$$dj = \frac{1}{3} \times \left(\frac{8}{9} + \frac{8}{9} + \frac{8-0}{8} \right) = 0.926$$

$$dw = 0.926 + (1 \times 0.1(1 - 0.926)) = 0.933$$

4. Kesalahan penulisan karena penukaran huruf

$$\text{String 1} = \text{dokumentsai} \quad s_1 = 11 \quad s_2 = 11 \quad m = 11$$

$$\text{String 2} = \text{dokumentasi} \quad t = 1 \quad l = 4 \quad p = 0.1$$

$$dj = \frac{1}{3} \times \left(\frac{11}{11} + \frac{11}{11} + \frac{11-1}{11} \right) = 0.969$$

$$dw = 0.969 + (4 \times 0.1(1 - 0.969)) = 0.981$$

3.4 Perhitungan Akurasi

Dilakukan perhitungan akurasi secara manual setelah pengujian sistem. Untuk mengetahui seberapa baik algoritma *Jaro-Winkler distance* mampu memperbaiki ejaan. Dapat dilihat rumus untuk menghitung akurasi pada Persamaan 3.1 di bawah ini:

$$Akurasi = \frac{jumlah\ kata\ yang\ benar}{jumlah\ kata} \times 100\% \quad 3.1$$

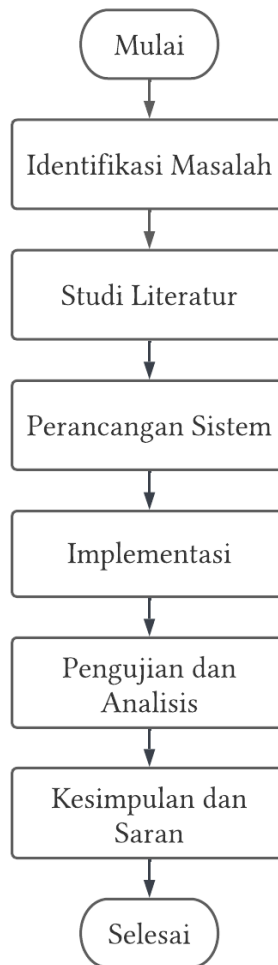
3.5 Skenario Pengujian

Layaknya sistem pada umumnya, guna menguji seberapa baik sistem dalam menjalankan fungsi-fungsi yang dimiliki yaitu mampu mengoreksi kata pada ejaan yang kurang tepat dan efisiensi waktu eksekusi sistem, maka dilakukan perancangan skenario pengujian. Pada penelitian ini terdapat dua skenario pengujian yang akan diterapkan. Skenario pengujian pertama untuk mengukur nilai akurasi algoritma *Jaro-Winkler Distance* dalam menentukan relevansi saran kata yang disesuaikan dengan kamus berdasarkan nilai *distance* yang terdekat. Pengujian dilakukan dengan membagi ke dalam empat jenis kesalahan penulisan yang berbeda, yaitu kesalahan yang disebabkan oleh penghapusan, penggantian, penambahan, dan penukaran huruf. Masing-masing data yang diuji berjumlah 40 sampel kata pada setiap jenis kesalahan yang diambil secara acak dari *dataset keyword* judul berita pariwisata.

Skenario pengujian kedua yakni mengukur waktu eksekusi pada sistem dari kedua jenis model pemrosesan yaitu serial dan paralel yang akan dilakukan menggunakan tiga perangkat keras dengan detail spesifikasinya dapat dilihat pada Tabel 1.1. Pengujian ini bertujuan untuk efisiensi sistem dengan membandingkan waktu eksekusi yang dibutuhkan pada pemrosesan secara serial (tanpa *multiprocessing*) maupun paralel (dengan *multiprocessing*) pada data uji berjumlah 4, 7, dan 10.

3.6 Tahapan Penelitian

Diagram alir mengenai langkah-langkah metode penelitian ditunjukkan pada Gambar 3.5.



Gambar 3.5 Diagram Alur Tahapan Penelitian

3.6.1 Identifikasi Masalah

Perumusan masalah adalah langkah awal dalam penelitian. Pada tahap ini dilakukan pencarian suatu fenomena untuk mendapatkan masalah yang ada, sehingga berdasarkan masalah yang diperoleh, dapat dilakukan penelitian untuk memecahkan permasalahan tersebut. Kemudian dilanjutkan untuk mempelajari dan menemukan kesimpulan dari hasil yang didapatkan. Permasalahan dalam penelitian ini adalah bagaimana hasil identifikasi kesalahan pengetikan teks dan saran perbaikan kata menggunakan algoritma *Jaro-Winkler Distance* dan optimasi efisiensi waktu pemrosesan menggunakan *multiprocessing*.

3.6.2 Studi Literatur

Studi Literatur adalah kumpulan referensi dan teori dari penelitian yang sudah dilakukan untuk dijadikan dasar dalam membantu melakukan penelitian ini.

Dilihat dari perannya yaitu untuk mengkaji penelitian sebelumnya agar literatur yang digunakan dapat memperbaiki kekurangan - kekurangan penelitian sebelumnya, maka studi literatur harus sesuai dengan permasalahan yang diangkat pada penelitian.

3.6.3 Perancangan Sistem

Studi literatur memperoleh hasil tentang apa saja yang diperlukan sistem sehingga dapat lanjut ke tahap berikutnya yaitu membuat perancangan sistem. Proses ini akan memaparkan tahapan perancangan arsitektur sistem dan *interface* aplikasi.

3.6.4 Implementasi

Setelah proses perancangan selesai, langkah selanjutnya adalah implementasi berdasarkan desain yang sudah dirancang sebelumnya. Pada sistem ini implementasi pembuatan sistem yang akan diaplikasikan adalah dengan menggunakan bahasa pemrograman *python*.

3.6.5 Pengujian dan Analisis

Tahapan pengujian sistem merupakan fase untuk melihat seberapa baik sistem yang dibuat, apakah sistem berjalan dengan baik sesuai yang direncanakan. Pengujian pertama untuk mengukur performa hasil koreksi ejaan kata algoritma *Jaro-Winkler Distance* pada empat jenis kesalahan penulisan, yaitu kesalahan yang disebabkan penambahan, penghapusan, penggantian, dan penukaran huruf. Pengujian kedua untuk mengukur perbandingan waktu pemrosesan sistem secara serial maupun paralel yang diujikan di tiga perangkat dengan spesifikasi yang berbeda. Setelah dilakukan pengujian sistem, hasilnya diolah dengan menghitung nilai akurasi dari koreksi kata yang dihasilkan oleh algoritma *Jaro-Winkler Distance* dan mengukur rasio perbandingan waktu eksekusi yang dibutuhkan pada kedua jenis model pemrosesan.

3.6.6 Kesimpulan dan Saran

Pada tahap ini dapat ditarik kesimpulan dari keseluruhan proses diawali dengan identifikasi masalah hingga tahap pengujian berdasarkan ketentuan yang telah ditetapkan sebelumnya. Secara umum, kesimpulan berisi hasil pengujian dan

analisis yang mengarah pada kesesuaian antara teori yang mendasari dengan penerapan kepada sistem. Begitu juga terdapat masukan dan saran yang diharapkan dapat menjadi bahan evaluasi untuk membangun sistem yang lebih baik.

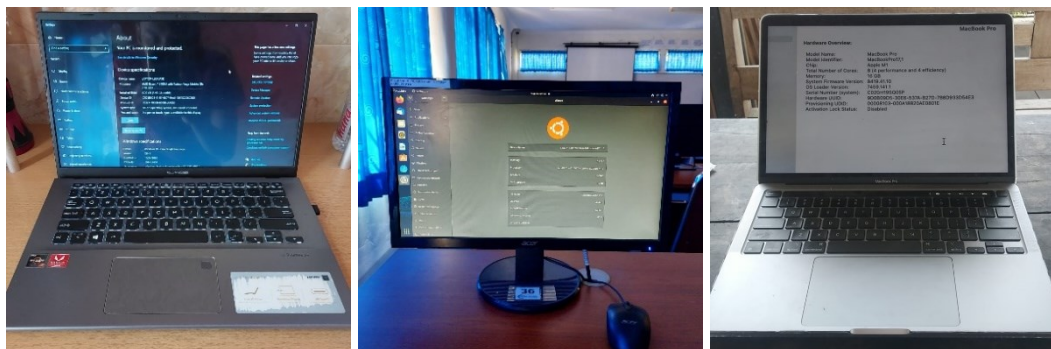
HASIL DAN PEMBAHASAN

Pada bab ini akan menjelaskan hasil dari uji coba dan pengimplementasian sistem berdasarkan perancangan dan metodologi yang telah dijelaskan pada bab sebelumnya. Pada pembahasan awal akan dijelaskan tahapan pembangunan sistem dimulai dari persiapan *tools* yang digunakan hingga pengujian sistem beserta analisis dan evaluasi yang dihasilkan.

4.1 Lingkungan Pengujian

Pada deskripsi lingkungan implementasi ini menjelaskan mengenai perangkat lunak dan perangkat keras yang digunakan dalam mengimplementasikan penelitian dari awal pengumpulan data hingga pengujian sistem.

Terdapat 3 perangkat keras yang digunakan dalam penelitian ini. Perangkat keras pertama yang dimiliki pribadi untuk semua aktivitas dalam penelitian ini. Perangkat keras kedua dan ketiga digunakan sebagai uji coba pengimplementasian *multiprocessing* untuk melihat perbedaan waktu eksekusi yang dihasilkan di beberapa perangkat dengan spesifikasi yang berbeda. Berikut gambar fisik beserta detail spesifikasi dari ketiga perangkat yang digunakan:



Gambar 4.1 Perangkat Keras Terpakai

a) Perangkat 1 (Asus VivoBook)

- CPU : AMD Ryzen 7 3700U
- *Core & Threads* : 4 cores & 8 threads
- RAM : 8 GB

- Sistem Operasi : Windows

b) Perangkat 2 (HP G4 MT Business PC)

- CPU : Intel i7-8700
- *Core & Threads* : 6 cores & 12 threads
- RAM : 8 GB
- Sistem Operasi : Linux Ubuntu

c) Perangkat 3 (Macbook Pro)

- CPU : Apple M1
- *Core & Threads* : 8 cores & 8 threads
- RAM : 16 GB
- Sistem Operasi : MacOS

Sistem yang dibuat pada penelitian ini menggunakan beberapa perangkat lunak dengan keterangan yang dijelaskan pada Tabel 4.1:

Tabel 4.1 Perangkat Lunak Terpakai

No.	Nama	Penjelasan
1.	Python 3.10	Sebuah bahasa pemrograman yang dipakai untuk implementasi algoritma. Python juga mempunyai banyak fungsi <i>built-in</i> yang dipakai dalam penelitian.
2.	phpMyAdmin	<i>Database server</i> berbasis web yang berfungsi untuk mengelola <i>database</i> MySQL.
3.	Xampp	<i>Web server</i> lintas <i>platform open-source</i> yang terdiri dari Apache HTTP Server, <i>database</i> MySQL, dan interpreter untuk skrip yang ditulis dalam bahasa pemrograman PHP dan Perl.
4.	VSCode	IDE (<i>Integrated Development Editor</i>) digunakan untuk meng- <i>edit file</i> pemrograman serta mengakses dan menjalankan aplikasi.
5.	Jupyter Notebook	Sebuah aplikasi digunakan untuk membuat <i>file</i> yang dapat menjalankan program dan menampilkan hasilnya secara runtut.

No.	Nama	Penjelasan
6.	Flask	<i>Web framework</i> yang ditulis dengan bahasa <i>python</i> sebagai kerangka kerja aplikasi dan tampilan dari web.
7.	Multiprocessing	<i>Package</i> atau <i>library</i> pada bahasa pemrograman <i>python</i> untuk membuat suatu proses dapat berjalan secara paralel.

4.2 Implementasi Dataset

Dataset yang digunakan dalam penelitian ini berupa daftar kata bahasa Indonesia yang terdiri dari 61.434 kosakata bahasa Indonesia dan juga 30.000 daftar kata bahasa Inggris yang paling umum digunakan. Keduanya digunakan sebagai acuan kata benar saat kata input dilakukan pengoreksian. Kemudian terdapat *dataset* berita pariwisata sebanyak 100 sebagai data yang ditampung untuk selanjutnya ditampilkan ketika *user* melakukan *request* pada sistem melalui *search engine*.

Terdapat juga *dataset* daftar kata *stopword* bahasa Indonesia dengan total 758 kata untuk menghapus kata yang dianggap tidak memiliki bobot atau dinilai tidak penting karena data *stopword* dianggap tidak berpengaruh terhadap kualitas hasil sistem. Selanjutnya di kolom judul pada *dataset* berita pariwisata dilakukan proses *preprocessing* yang menghasilkan *term list* sebanyak 722 kata unik. Langkah ini dilakukan untuk menambahkan kata yang belum terdapat pada *dataset* kamus bahasa agar saran kata yang dihasilkan sistem setelah melalui proses pengoreksian bisa maksimal dan menyeluruh terhadap data yang ada. *Dataset* yang terakhir yakni daftar nama tempat dengan jumlah sebanyak 54 kata yang diambil secara manual dari kolom judul pada *dataset* berita pariwisata. Semua data tersebut disimpan dalam format *excel* yang kemudian di-*convert* ke dalam bentuk MySQL sebagai *database*.

4.3 Implementasi Sistem

Pada implementasi sistem ini akan membahas sistem seperti apa yang akan dibangun nantinya dalam menyelesaikan suatu permasalahan koreksi ejaan kata menggunakan *Jaro-Winkler Distance* pada sistem pencarian berita pariwisata yang dioptimasi waktu eksekusinya menggunakan *multiprocessing*. Sebelum data input

dikoreksi ejaan katanya, tentunya harus melewati proses *preprocessing* terlebih dahulu. Setelah proses *preprocessing* selesai dilakukan maka akan masuk ke proses berikutnya yakni pengecekan apakah termasuk kata baku, jika tidak maka akan dilakukan proses perbaikan kata menggunakan algoritma *Jaro-Winkler Distance* dengan cara mencocokkan setiap karakter *string* dari kata tidak baku dengan kata acuan yang diambil dari kamus pada *database*. Kemudian sistem akan dioptimasi waktu pemrosesannya menggunakan *multiprocessing* dengan membuat sistem berjalan secara paralel dengan cara membagi setiap kata pada data input ke dalam bagian proses tersendiri.

Hasil keluaran sistem yang akan dibangun ini berupa hasil perbaikan kata dari masing-masing data input yang diberikan pada mesin pencarian berita pariwisata, kemudian dioptimasi waktu eksekusinya menggunakan model pemrosesan secara paralel dengan penerapan *multiprocessing*.

4.3.1 *Preprocessing*

Fungsi dilakukannya tahapan *preprocessing* ini adalah untuk mengekstrak data teks yang semula tidak terstruktur menjadi data teks yang lebih terstruktur, sehingga sistem dapat lebih mudah dalam menjalankan proses lanjutan setelah *preprocessing* ini. Tahapan *preprocessing* melalui beberapa langkah dimulai dari yang pertama yaitu *cleaning*, selanjutnya *case folding*, kemudian *tokenizing*, dan tahap terakhir adalah *stopword removal*. Namun *preprocessing* yang digunakan pada sistem ini tidak melalui proses *stemming*, karena dalam proses pengubahan menjadi kata dasar akan menghilangkan imbuhan, sedangkan untuk proses identifikasi kesalahan kata dibutuhkan kata yang sama persis tanpa terdapat adanya perubahan dari kata aslinya. Hasil keluaran dari tahap ini yakni berupa daftar kata (*term list*) yang mewakili setiap data masukan yang diberikan oleh *user* pada *search engine*. Detail *listing program preprocessing* dapat dilihat pada Kode Program 4.1.

Kode Program 4.1 *Preprocessing Text*

1	<code>def preprocessingtext(kalimat, daftar_kata_stopword):</code>
2	
3	<code> lower_case = kalimat.lower()</code>
4	
5	<code> kalimat = lower_case.strip()</code>
6	

7	punctuation = '!"()-[]{};:'"\ ,.<>/?@#\$\$%^&*~''
8	cleaning_punctuation = ""
9	for char in kalimat:
10	if (char not in punctuation):
11	cleaning_punctuation = cleaning_punctuation +
12	char
13	
14	cleaning_number = ""
15	for char in cleaning_punctuation:
16	if not char.isdigit():
17	cleaning_number = cleaning_number + char
18	
19	tokens = cleaning_number.split()
20	
21	stp = []
22	for i in tokens:
23	if i not in daftar_kata_stopword:
24	stp.append(i)
25	
26	return stp
27	

Penjelasan Kode Program 4.1 *Preprocessing Text*:

- Baris 1 : membuat *function* untuk *preprocessing text* dengan 2 parameter yakni kalimat yang diinputkan dan daftar kata *stopword* yang diambil dari *database*.
- Baris 3 : *case folding* data kalimat dengan menggunakan *function* `lower()`
- Baris 5 : *cleaning whitespace* pada data kalimat dengan menggunakan *function* `strip()`
- Baris 7 - 12 : *cleaning* simbol dan tanda baca yang tidak dibutuhkan oleh sistem pada data kalimat
- Baris 14 - 17 : *cleaning* angka pada data kalimat
- Baris 19 : tokenisasi data kalimat dengan menggunakan *function* `split()`
- Baris 21 - 24 : *stopword removal* pada data kalimat dengan data kata *stopword* yang diambil dari *database*.
- Baris 26 : Mengembalikan nilai pada variabel `stp` ketika *function* `preprocessingtext` dipanggil.

4.3.2 Jaro-Winkler Distance

Algoritma *Jaro-Winkler Distance* digunakan untuk memperbaiki kata yang tidak baku atau terdapat kesalahan ketik. Perbaikan kata yang dimaksudkan yakni mengukur kemiripan pada dua *string* yaitu antara *string* data input dan *string* acuan dari *database*, jika nilai *distance* semakin tinggi atau mendekati dengan 1 maka *string* tersebut semakin mirip dengan *string* acuan yang terdapat pada kamus. Proses ini dibagi menjadi sejumlah tahapan, yang pertama menghitung panjang *string*, menghitung jumlah karakter yang sama pada kedua *string*, dan mencari jumlah transposisi karakter pada *string*. Hasil keluaran dari tahap ini adalah perubahan dari kata tidak baku menjadi kata baku berdasarkan Kamus Besar Bahasa Indonesia. Detail *listing program* algoritma dapat dilihat pada Kode Program 4.2.

Kode Program 4.2 Algoritma Jaro-Winkler Distance

```
1  from math import floor
2
3  def jaroDistance(string1, string2):
4
5      if (string1 == string2):
6          return 1.0
7
8      len_string1 = len(string1)
9      len_string2 = len(string2)
10
11     max_distance = floor(max(len_string1, len_string2) / 2)
12 - 1
13
14     match = 0
15
16     match_string1 = [0] * len(string1)
17     match_string2 = [0] * len(string2)
18
19     for i in range(len_string1):
20
21         for j in range(max(0, i - max_distance),
22 min(len_string2, i + max_distance + 1)):
23
24             if (string1[i] == string2[j] and
25 match_string2[j] == 0):
26                 match_string1[i] = 1
27                 match_string2[j] = 1
28                 match += 1
29                 break
30
31     if (match == 0):
32         return 0.0
33
```

```

34     transposition = 0
35     pointer = 0
36
37     for i in range(len_string1):
38
39         if (match_string1[i]):
40
41             while (match_string2[pointer] == 0):
42                 pointer += 1
43
44             if (string1[i] != string2[pointer]):
45                 t += 1
46
47             pointer += 1
48
49     transposition = transposition // 2
50
51     return (match / len_string1 + match / len_string2 +
52            (match - transposition) / match) / 3.0
53
54 def jaroWinklerDistance(string1, string2):
55
56     jaro_distance = jaroDistance(string1, string2)
57
58     if (jaro_distance > 0.7):
59
60         prefiks = 0
61         for i in range(min(len(string1), len(string2))):
62
63             if (string1[i] == string2[i]):
64                 prefiks += 1
65
66             else:
67                 break
68
69         prefiks = min(4, prefiks)
70
71         jaro_distance += 0.1 * prefiks * (1 - jaro_distance)
72
73     return jaro_distance

```

Penjelasan Kode Program 4.2 Algoritma *Jaro-Winkler Distance*:

- Baris 1 : *import* floor dari modul math untuk membulatkan bilangan desimal ke bawah
- Baris 3 : membuat *function* jaroDistance dengan 2 parameter yaitu *string1* sebagai data input dan *string2* sebagai kata kamus dari *database*
- Baris 5 - 6 : pengecekan jika *string1* dan *string2* sama maka langsung mengembalikan nilai 1

- Baris 8 - 9 : menampung nilai panjang dari *string1* dan *string2*
- Baris 11 - 12 : variabel yang menampung rumus *max distance* karakter dapat disamakan
- Baris 14 : deklarasi variabel `match` dengan nilai 0 yang digunakan untuk menampung jumlah karakter yang sama
- Baris 16 - 17 : membuat variabel *list* kosong dengan panjang sesuai panjang karakter dari *string1* dan *string2*
- Baris 19 - 29 : iterasi untuk mencari karakter yang sama sesuai dengan nilai *max distance*
- Baris 31 - 32 : pengecekan nika variabel `match` bernilai 0 maka akan mengembalikan nilai 0
- Baris 34 - 35 : deklarasi variabel `transposition` dengan nilai awal 0 untuk menampung jumlah kata transposisi. Variabel `pointer` dengan nilai awal 0 untuk penanda karakter yang dicek pada *string2*
- Baris 37 - 49 : iterasi untuk mencari jumlah karakter transposisi pada *string1* dengan *string2* sebagai pembanding dan kemudian dibagi 2
- Baris 51 - 52 : mengembalikan nilai dari hasil perhitungan rumus *Jaro Distance* (dj) ketika *function* `jaroDistance` dipanggil
- Baris 54 : membuat *function* `jaroWinklerDistance` dengan 2 parameter yaitu *string1* sebagai data input dan *string2* sebagai kata kamus dari *database*
- Baris 56 : inisialisasi variabel yang menampung nilai dari *function* `jaroDistance()`
- Baris 58 - 67 : pengecekan nilai pada variabel `jaro_distance` jika melebihi 0.7 maka akan dilakukan iterasi untuk mencari jumlah prefiks
- Baris 69 : variabel untuk mengambil nilai prefiks dengan mengambil angka terendah dari variabel `prefiks` atau nilai 4 sebagai batas maksimal nilai prefiks pada rumus *Jaro-Winkler Distance*
- Baris 71 : menghitung hasil dari rumus *Jaro-Winkler Distance* (dw)
- Baris 73 : mengembalikan nilai dari variabel `jaro_distance` ketika *function* `jaroWinklerDistance` dipanggil

4.3.3 Multiprocessing

Pada tahap *multiprocessing* dilakukan pengoptimalan waktu eksekusi sistem dengan membagi setiap kata pada data input ke dalam proses tersendiri secara paralel dari yang sebelumnya secara serial atau kata diproses dengan cara satu persatu.

Proses dimulai dengan *import* seluruh *library* dan *file* yang dibutuhkan termasuk *library multiprocessing* untuk melakukan proses secara paralel. Selanjutnya konfigurasi *database* dan sesuaikan dengan nama *host*, *user*, *password*, dan nama *database* sesuai pada perangkat yang digunakan. Setelahnya instansiasi *object flask* dan *set secret key* dengan nilai acak yang wajib diberikan ketika menggunakan *session*. *Session* digunakan menyimpan informasi yang dikirimkan *user* untuk di simpan di *server* agar kemudian nilainya dapat dipanggil di halaman lain. Langkah utama pada proses ini yakni mengaktifkan jalur proses paralel sebanyak jumlah CPU pada perangkat. Kemudian mendistribusikan data daftar kata kamus pada setiap *pool* atau proses. Lalu masing-masing kata pada data input disebarkan pada setiap jalur proses untuk dilakukan proses koreksi ejaan kata. Perhitungan waktu eksekusinya sendiri dimulai ketika *user* melakukan *request* dengan mengirimkan *keyword* di mesin pencarian sampai dengan menampilkan hasil data berita pariwisata. Detail *listing program multiprocessing* dapat dilihat pada Kode Program 4.3.

Kode Program 4.3 Multiprocessing

```
1  from flask import Flask, render_template, request, session
2  from flask_mysql import MySQL
3  from jarowinklerdistance import *
4  from preprocessing import *
5  from multiprocessing import Pool, CPU_count
6  import time
7
8  app = Flask(__name__)
9  app.secret_key =
10  "#$%#$%^%BFGBFGBSFGNSGJTNADFHH@#%$%#T#FFWF$^F@$F#$FW"
11
12  mysql = MySQL(app)
13  daftar_kata = None
14  daftar_kata_stopword = None
15  pool_worker = None
16
```

```

17 def load_daftar_kata():
18     global daftar_kata
19     if daftar_kata is not None:
20         return
21
22     cur = mysql.connection.cursor()
23     cur.execute("SELECT * FROM daftar_kata")
24     daftar_kata = [x[1] for x in cur.fetchall()]
25
26 def load_daftar_kata_stopword():
27     global daftar_kata_stopword
28     if daftar_kata_stopword is not None:
29         return
30
31     cur = mysql.connection.cursor()
32     cur.execute("SELECT stopwords_wordlist FROM
33 kata_stopword")
34     daftar_kata_stopword = [x[0] for x in cur.fetchall()]
35
36 def suggest_fix(search):
37     global daftar_kata
38     list_jwd_value = {}
39
40     for word in daftar_kata:
41         list_jwd_value[word] = jaroWinklerDistance(search,
42 word)
43
44     kata_koreksi = max(list_jwd_value,
45 key=list_jwd_value.get)
46
47     return kata_koreksi
48
49 def init_worker(shared_daftar_kata):
50     global daftar_kata
51     daftar_kata = shared_daftar_kata
52
53 @app.route('/')
54 def index():
55     cur = mysql.connection.cursor()
56     cur.execute("SELECT * FROM berita_pariwisata")
57     data = cur.fetchall()
58     cur.close()
59
60     return render_template('index.html',
61 beritapariwisata=data)
62
63 def getberita(search: list[str]):
64     cursor = mysql.connection.cursor()
65     cursor.execute(
66         "SELECT berita_pariwisata.* FROM keyword_berita,
67 berita_pariwisata, daftar_kata WHERE
68 berita_pariwisata.id_berita = keyword_berita.kata_berita
69 AND daftar_kata.id_kata = keyword_berita.kata_input AND
70

```

```

71 daftar_kata.wordlist IN ({})) GROUP BY
72 berita_pariwisata.id_berita HAVING count(*) = {}".format(
73     ",".join(['{}' + i + '{}' for i in search]),
74     len(search))
75 )
76 results = cursor.fetchall()
77 cursor.close()
78 return results
79
80 @app.route("/result", methods=["GET", "POST"])
81 def result():
82     if request.method == "POST":
83         global pool_worker
84         load_daftar_kata()
85         load_daftar_kata_stopword()
86         if not pool_worker:
87             pool_worker = Pool(CPU_count(), init_worker,
88 [daftar_kata])
89             start = time.time()
90             data = dict(request.form)
91             search = data["search"]
92             sugesti = []
93             search = [x for x in preprocessingtext(search,
94 daftar_kata_stopword)]
95             sugesti = pool_worker.map(suggest_fix, search)
96             berita = getberita(search)
97             session["search"] = ' '.join(search)
98             sugesti = ' '.join(sugesti)
99             end = time.time()
100            waktu = round(end-start,2)
101        else:
102            berita = []
103
104        return render_template("result.html", hasilcari=berita,
105 search=session["search"], waktu=waktu, sugesti=sugesti)
106
107 if __name__ == "__main__":
108     app.run(debug=True)

```

Penjelasan Kode Program 4.3 *Multiprocessing*:

- Baris 1 : *import library* flask dan panggil *object* Flask, *render_template* untuk *render file*, *request* untuk menerima masukan dari *form* secara *POST/GET*, dan *session* untuk menyimpan informasi yang dikirimkan *user* untuk di simpan di *server*
- Baris 2 - 6 : *import library* lainnya yang dibutuhkan seperti *flask_mysqldb*, *multiprocessing*, dan *time*. Selain itu

import semua nilai yang terdapat di *file preprocessing* dan *jarowinklerdistance*

- Baris 8 - 10 : inialisasi variabel `app` yang berisikan *object* Flask kemudian buat *secret key* dengan random teks yang wajib digunakan ketika menggunakan *session*
- Baris 12 - 15 : inialisasi variabel `mysql` yang berisikan *object* MySQL. Dibuat juga variabel `pool_worker`, `daftar_kata`, `daftar_kata_stopword`, dengan nilai awal *None*. Langkah ini dilakukan sebagai *trigger* proses *load* data ketika dipanggil pada *function* lain
- Baris 17 - 24 : membuat *function* `load_daftar_kata` untuk melakukan *query* daftar kata. Pada *function* ini panggil variabel global yang berada di luar cakupan lingkungan fungsi yang telah dideklarasikan baris *code* sebelumnya. Kemudian pada baris 24 - 25 dilakukan pengecekan jika *statement* bernilai *true* maka data ditampilkan. Jika tidak maka baris 27 - 29 dijalankan untuk melakukan *query* daftar kata pada *database*
- Baris 26 - 34 : membuat *function* `load_daftar_kata_stopword` untuk melakukan *query* daftar kata *stopword*. Pada *function* ini panggil variabel global yang berada di luar cakupan lingkungan fungsi yang telah dideklarasikan baris *code* sebelumnya. Kemudian pada baris 33 - 34 dilakukan pengecekan jika *statement* bernilai *true* maka data ditampilkan. Jika tidak maka baris 36 - 39 dijalankan untuk mengambil data dengan melakukan *query* daftar kata *stopword* pada *database*
- Baris 36 - 47 : membuat *function* `suggest_fix` dengan parameter *search* yaitu data input yang diberikan pada mesin pencarian. *Function* ini dibuat untuk menghitung nilai kedekatan *distance* pada tiap kata pada data input dengan kamus dan mengembalikan kata dengan nilai *distance* terbesar (terdekat) dengan menggunakan fungsi `max`
- Baris 49 - 51 : membuat *function* `init_worker` dengan parameter data daftar kata. *Function* ini dibuat untuk menyebarkan data daftar kata pada setiap jalur paralel

- Baris 53 - 61 : merutekan *flask* pada halaman `index.html`. Kemudian buat *function* `index` untuk menulis *statement* apa saja yang ingin dilakukan pada halaman tersebut. Di sini dilakukan *query* terhadap data berita pariwisata, selanjutnya di-render dengan nama `beritapariwisata` ketika dipanggil pada *file* `index.html`
- Baris 63 - 77 : membuat *function* `get_berita` dengan parameter *search* yaitu data input yang diberikan pada mesin pencarian dengan tipe data *list*. Kemudian dilakukan iterasi pada tiap kata input dengan *query*. *Function* ini dibuat untuk mendapatkan data berita pariwisata berdasarkan data input yang diberikan
- Baris 79 - 104 : merutekan *flask* pada halaman `result.html`. Kemudian buat *function* `result` untuk menulis *statement* apa saja yang ingin dilakukan pada halaman tersebut ketika terdapat *request POST* maka dilakukan proses koreksi ejaan kata dengan membagi data input pada setiap *track* proses sebanyak jumlah CPU. Terdapat juga proses untuk menghitung waktu eksekusi ketika *request* dikirimkan sampai dengan menampilkan hasil
- Baris 106 - 107 : menjalankan *flask* dan `debug=True` untuk menyimpan secara otomatis ketika terdapat perubahan

4.4 Pengujian dan Analisis

Seperti halnya suatu sistem pada umumnya setelah melalui proses perancangan sistem dan pembuatan sistem, maka langkah selanjutnya adalah menguji sistem tersebut. Tujuan dari tahap pengujian adalah untuk mengetahui apakah sistem yang dibuat memiliki fungsionalitas yang baik atau belum, selain itu pengujian juga digunakan untuk menguji kesesuaian nilai yang dihasilkan baik dengan perhitungan manual dan yang diterapkan pada sistem. Pengujian yang akan dilakukan terdiri atas pengujian untuk mengukur performa hasil koreksi ejaan kata algoritma *Jaro-Winkler Distance* pada 4 jenis kesalahan penulisan yaitu penambahan, penghapusan, penggantian, dan penukaran huruf. Selanjutnya dilakukan pengujian untuk mengukur perbandingan waktu pemrosesan sistem secara serial maupun paralel dengan penerapan *multiprocessing* pada data uji berjumlah 4, 7, dan 10.

4.4.1 Hasil Skenario Pengujian Pertama

Skenario pengujian pertama dilakukan untuk mengukur performa algoritma *Jaro-Winkler Distance* dalam menentukan relevansi saran kata sesuai kamus berdasarkan nilai *distance* yang dihasilkan dengan menghitung nilai akurasi dari setiap jenis kesalahan yang terdiri dari kesalahan yang disebabkan penambahan, penghapusan, penggantian, dan penukaran huruf. Aplikasi ini menggunakan pendekatan *user relevance judgement* dalam menentukan relevan atau tidaknya saran kata yang dihasilkan algoritma pada sistem. Masing-masing data yang diuji berjumlah 40 sampel kata pada setiap jenis kesalahan yang diambil secara acak dari *dataset keyword* judul berita pariwisata.

Kolom kata salah pada tabel adalah kemungkinan terjadinya penulisan kata salah ketik atau *typo*, kolom kata benar mereferensikan kata baku pada kamus, dan kolom hasil sistem adalah output yang dikeluarkan oleh sistem sesuai perhitungan dari algoritma *Jaro-Winkler Distance*. Nilai 0 pada kolom hasil validasi mengartikan hasil kata salah yang dikoreksi oleh sistem tidak merujuk kepada kata benar yang dimaksudkan, dan nilai 1 menandakan sama persis dengan yang direferensikan pada kolom kata benar. Dalam penelitian ini terdapat 3 *user relevance judgement* termasuk dengan peneliti di dalamnya yang terlibat menentukan saran kata yang muncul relevan atau tidak. Detail keseluruhan hasil *user relevance judgement* dapat dilihat pada Lampiran 2.

4.4.1.1 Penambahan Huruf

Pada Tabel 4.2 pengujian data uji berupa adanya penambahan karakter huruf di mana terdapat karakter lebih dari yang seharusnya terdapat pada kamus.

Tabel 4.2 Pengujian Kesalahan Penulisan Penambahan Huruf

No.	Kata Salah	Kata Benar	Hasil Sistem	Hasil Validasi
1.	wisatta	wisata	wisata	1
2.	panatai	pantai	panai	0
3.	inndah	indah	indah	1
4.	hutana	hutan	hutan	1
5.	bukkit	bukit	bukit	1
6.	pullau	pulau	pulau	1

No.	Kata Salah	Kata Benar	Hasil Sistem	Hasil Validasi
7.	gunnung	gunung	gunung	1
8.	panaorama	panorama	panorama	1
9.	tikket	tiket	tiket	1
10.	suneset	sunset	sunset	1
11.	rekomenedasi	rekomendasi	rekomendasi	1
12.	waduuk	waduk	waduk	1
13.	kuotta	kuota	kuota	1
14.	bnuka	buka	buka	1
15.	possko	posko	posko	1
16.	liburran	liburan	liburan	1
17.	monummen	monumen	monumen	1
18.	musieum	museum	museum	1
19.	peSSona	pesona	pesona	1
20.	populler	populer	populer	1
21.	tebbing	tebing	tebing	1
22.	suassana	suasana	suasana	1
23.	desitinasi	destinasi	desis	0
24.	allam	alam	alam	1
25.	religih	religi	religi	1
26.	kabhupaten	kabupaten	kabupaten	1
27.	virral	viral	viral	1
28.	rutte	rute	rute	1
29.	lokkasi	lokasi	lokasi	1
30.	informassi	informasi	informasi	1
31.	favorrit	favorit	favorit	1
32.	legghenda	legenda	legenda	1
33.	ekksotis	eksotis	eksotis	1
34.	manggrove	mangrove	mangrove	1
35.	dunnia	dunia	dunia	1
36.	nassional	nasional	nasional	1
37.	kerraton	keraton	keraton	1
38.	hargha	harga	harga	1
39.	fasillitas	fasilitas	fasilitas	1
40.	massuk	masuk	masuk	1
Akurasi				0.95

Seperti yang tertera Tabel 4.2 hasil pengujian kesalahan penulisan yang disebabkan adanya penambahan karakter huruf menghasilkan akurasi sebesar 0.95.

4.4.1.2 Penghapusan Huruf

Pada Tabel 4.3 pengujian data uji berupa adanya penghapusan karakter di mana terdapat pengurangan karakter huruf dari yang seharusnya pada kamus.

Tabel 4.3 Pengujian Kesalahan Penulisan Penghapusan Huruf

No.	Kata Salah	Kata Benar	Hasil Sistem	Hasil Validasi
1.	wsata	wisata	wisata	1
2.	pntai	pantai	pantai	1
3.	inda	indah	indah	1
4.	utan	hutan	utang	0
5.	buki	bukit	bukit	1
6.	pulu	pulau	puluh	0
7.	gunug	gunung	gunung	1
8.	panrama	panorama	panorama	1
9.	tiet	tiket	tiket	1
10.	snset	sunset	sunset	1
11.	rekomedasi	rekomendasi	rekomendasi	1
12.	wduk	waduk	waduk	1
13.	kuot	kuota	kuota	1
14.	uka	buka	ukas	0
15.	posk	posko	posko	1
16.	libuan	liburan	liburan	1
17.	mnumen	monumen	monumen	1
18.	museu	museum	museum	1
19.	psona	pesona	pesona	1
20.	pouler	populer	populer	1
21.	tbing	tebing	tebing	1
22.	suasna	suasana	suasana	1
23.	detinasi	destinasi	destinasi	1
24.	alm	alam	alam	1
25.	reigi	religi	religi	1
26.	kaupaten	kabupaten	kabupaten	1
27.	virl	viral	viral	1
28.	rte	rute	rote	0
29.	lokai	lokasi	lokasi	1
30.	iformasi	informasi	informasi	1
31.	favori	favorit	favorit	1

No.	Kata Salah	Kata Benar	Hasil Sistem	Hasil Validasi
32.	legeda	legenda	legenda	1
33.	esotis	eksotis	eksotis	1
34.	mangrov	mangrove	mangrove	1
35.	dnia	dunia	dunia	1
36.	nasinal	nasional	nasional	1
37.	kerton	keraton	keraton	1
38.	harg	harga	harga	1
39.	fasilita	fasilitas	fasilitas	1
40.	msuk	masuk	masuk	1
Akurasi				0.90

Seperti tertera Tabel 4.3 hasil pengujian kesalahan penulisan yang disebabkan adanya penghapusan karakter huruf menghasilkan nilai akurasi sebesar 0.90.

4.4.1.3 Penggantian Huruf

Pada Tabel 4.4 pengujian data yang diujikan berupa adanya penggantian karakter huruf di mana terdapat huruf yang berbeda dari yang seharusnya terdapat pada kamus.

Tabel 4.4 Pengujian Kesalahan Penulisan Penggantian Huruf

No.	Kata Salah	Kata Benar	Hasil Sistem	Hasil Validasi
1.	wisats	wisata	wisata	1
2.	pantao	pantai	pamit	0
3.	indsh	indah	indah	1
4.	hutam	hutan	hutan	1
5.	bikit	bukit	biskuit	0
6.	oulau	pulau	pulau	1
7.	gunumg	gunung	gunung	1
8.	pamorama	panorama	panorama	1
9.	tijet	tiket	tiket	1
10.	sumset	sunset	sunset	1
11.	rekomensasi	rekomendasi	rekomendasi	1
12.	wadik	waduk	waduk	1
13.	kiota	kuota	kita	0

No.	Kata Salah	Kata Benar	Hasil Sistem	Hasil Validasi
14.	buja	buka	bujal	0
15.	oosko	posko	otoskop	0
16.	libiran	liburan	liburan	1
17.	mpnumen	monumen	monumen	1
18.	museun	museum	museum	1
19.	oesona	pesona	pesona	1
20.	popiler	populer	populer	1
21.	tebing	tebing	tebing	1
22.	auasana	suasana	alasan	0
23.	desrinasi	destinasi	destinasi	1
24.	alqm	alam	alam	1
25.	rekigi	religi	religi	1
26.	ksbupsten	kabupaten	kabupaten	1
27.	virak	viral	irak	0
28.	rite	rute	ritme	0
29.	lpkasi	lokasi	lokasi	1
30.	imformasi	informasi	informasi	1
31.	faforit	favorit	favorit	1
32.	leyenda	legenda	legenda	1
33.	exsotis	eksotis	eksotis	1
34.	mangrofe	mangrove	mangrove	1
35.	dumia	dunia	dumi	0
36.	nasionsl	nasional	nasional	1
37.	keraron	keraton	keran	0
38.	harha	harga	hara	0
39.	fasikitas	fasilitas	fasilitas	1
40.	mssuk	masuk	masuk	1
Akurasi				0.725

Seperti yang tertera Tabel 4.4 hasil pengujian kesalahan penulisan yang disebabkan adanya penggantian karakter huruf menghasilkan nilai akurasi sebesar 0.725.

4.4.1.4 Penukaran Huruf

Pada Tabel 4.5 pengujian data uji berupa adanya penukaran karakter huruf di mana terdapat huruf yang posisinya tertukar dengan karakter huruf lain dari yang seharusnya pada kamus.

Tabel 4.5 Pengujian Kesalahan Penulisan Penukaran Huruf

No.	Kata Salah	Kata Benar	Hasil Sistem	Hasil Validasi
1.	wiasta	wisata	wisata	1
2.	pantia	pantai	pantai	1
3.	indha	indah	indah	1
4.	huatn	hutan	hutan	1
5.	bkuit	bukit	bukit	1
6.	pulua	pulau	pula	0
7.	gunugn	gunung	gunung	1
8.	panoraam	panorama	panorama	1
9.	itket	tiket	tiket	1
10.	sunste	sunset	sunset	1
11.	reokmenadsi	rekomendasi	rekomendasi	1
12.	wadku	waduk	waduk	1
13.	kouta	kuota	kota	0
14.	buak	buka	buka	1
15.	opsko	posko	posko	1
16.	liubran	liburan	liburan	1
17.	monumne	monumen	monumen	1
18.	musuem	museum	museum	1
19.	peosna	pesona	pesona	1
20.	populre	populer	populer	1
21.	teibng	tebing	tebing	1
22.	sausana	suasana	sasana	0
23.	destiansi	destinasi	destinasi	1
24.	aalm	alam	alam	1
25.	reilgi	religi	religi	1
26.	kbaupatne	kabupaten	kabupaten	1
27.	ivral	viral	viral	1
28.	rtue	rute	rute	1
29.	loaksi	lokasi	lokasi	1
30.	inforamsi	informasi	informasi	1
31.	favoirt	favorit	favorit	1
32.	legedna	legenda	legenda	1
33.	ekostis	eksotis	eksotis	1
34.	mangroev	mangrove	mangrove	1
35.	dunai	dunia	dunia	1
36.	nasoinla	nasional	nasional	1

No.	Kata Salah	Kata Benar	Hasil Sistem	Hasil Validasi
37.	kearton	keraton	keraton	1
38.	hagra	harga	hara	0
39.	faislitas	fasilitas	fasilitas	1
40.	amsuk	masuk	amuk	0
Akurasi				0.875

Seperti yang tertera Tabel 4.5 hasil pengujian kesalahan penulisan disebabkan huruf yang posisinya tertukar dengan karakter huruf lain menghasilkan akurasi sebesar 0.875.

4.4.1.5 Analisis Hasil Skenario Pengujian Pertama

Hasil dari pengujian pertama yang telah dilakukan pada 40 kata pada setiap jenis kesalahan tertera pada Tabel 4.6.

Tabel 4.6 Hasil Pengujian Pertama

	Jenis Kesalahan				Rata-Rata
	Hapusi	Tambahi	Gantii	Tukari	
Akurasi	0.95	0.90	0.725	0.875	0.86

Berdasarkan Tabel 4.6 di atas menunjukkan bahwa algoritma *Jaro-Winkler Distance* menghasilkan akurasi yang berbeda pada setiap jenis kesalahan penulisan. Nilai akurasi tertinggi terdapat pada jenis kesalahan penghapusan huruf dengan akurasi sebesar 0.95 dan akurasi terendah pada jenis kesalahan penggantian huruf dengan nilai akurasi 0.725. Sedangkan pada jenis kesalahan penambahan dan penukaran huruf masing-masing menghasilkan nilai akurasi sebesar 0.90 dan 0.875. Berdasarkan pengujian pertama ini, dapat disimpulkan bahwa *Jaro-Winkler Distance* mampu mengoreksi dan memberikan saran kata pada kata yang kurang tepat dengan rata-rata akurasi pada semua jenis kesalahan dengan nilai akurasi sebesar 0.86.

4.4.2 Hasil Skenario Pengujian Kedua

Skenario pengujian kedua dilakukan untuk mengukur waktu eksekusi dari kedua jenis model pemrosesan yang dilakukan di beberapa *hardware* dengan

spesifikasi yang berbeda. Pengujian ini bertujuan untuk membandingkan waktu eksekusi yang dibutuhkan sistem pada pemrosesan secara serial (tanpa *multiprocessing*) maupun paralel (dengan *multiprocessing*) pada data kata uji berjumlah 4, 7, dan 10.

4.4.2.1 Analisis Hasil Skenario Pengujian Kedua

Pada pengujian ini *query* data uji berupa kata dari judul pada data berita pariwisata yang dibuat *typo* atau terdapat kesalahan ketik di setiap kata untuk diuji waktu pemrosesan yang dibutuhkan sistem dalam mengoreksi kata uji tersebut. Berikut *query* kata uji yang digunakan dalam pengujian dapat dilihat pada Tabel 4.7.

Tabel 4.7 Kata Uji Pada Pengujian Kedua

Jumlah Kata	Kata Uji
4	detinasi wisata kbaupatne sumene
7	desrinasi wsata allam ksbupsten suemnep jawwa tmur
10	destiansi wisats aalm bukkit daramitsa twrkwnal kaupaten sumennep jjawa tiimur

Hasil dari pengujian dari kata uji pada Tabel 4.7 yang telah dilakukan pada tiga perangkat yang berbeda, tertera pada Tabel 4.8.

Tabel 4.8 Hasil Pengujian Kedua

Jumlah Kata	Waktu Eksekusi (detik)					
	Serial			Paralel		
	Asus VivoBook (Windows)	HP (Linux Ubuntu)	MacBook Pro (MacOS)	Asus VivoBook (Windows)	HP (Linux Ubuntu)	MacBook Pro (MacOS)
4	4.56	1.89	1.76	1.92	0.86	0.61
7	6.62	2.64	2.58	2.10	0.95	0.69
10	10.31	4.35	4.03	3.16	1.19	1.02

Berdasarkan Tabel 4.8 di atas menunjukkan bahwa model pemrosesan secara paralel mampu mempercepat waktu eksekusi pada setiap perangkat yang diujikan. Waktu eksekusi terpendek terdapat pada perangkat MacBook Pro baik

pada proses secara serial maupun paralel dengan rasio perbandingan 1.76 : 0.61 detik pada jumlah kata uji sebanyak 4.

4.5 Tampilan Antarmuka

Bagian ini menjelaskan pembuatan *user interface* aplikasi berbasis web yang dirancang dengan sederhana sehingga memungkinkan *user* dapat menggunakan secara mudah. Aplikasi ini mampu membantu *user* dalam mengoreksi kesalahan ejaan kata pada *keyword* yang diberikan di dalam mesin pencarian untuk menemukan berita pariwisata yang sedang ingin dicari.

4.5.1 Tampilan Awal Aplikasi

Mulanya aplikasi akan menampilkan halaman *dashboard* pada saat pertama kali diakses. Halaman *dashboard* tersebut berisikan deskripsi singkat terkait aplikasi dan *search engine* untuk mencari berita pariwisata yang diinginkan. Terdapat juga daftar berita pariwisata sebagai sampel yang ditunjukkan pada halaman aplikasi. Bentuk tampilan visual *dashboard* dari aplikasi yang telah dibuat dapat dilihat pada Gambar 4.2.



Gambar 4.2 Tampilan Dashboard

4.5.2 Tampilan Hasil Pencarian

Halaman selanjutnya pada aplikasi yakni menampilkan hasil pencarian berdasarkan *keyword* yang diberikan pada sistem. Berikut hasil tampilan visual aplikasi dapat dilihat pada Gambar 4.3.

Portal Berita Pariwisata Madura

Waktu eksekusi: 4.43

Apakah yang anda maksud: destinasi wisata kabupaten sumenep?

No search results.

180411100135 - Muhammad Ali Yusuf

Gambar 4.3 Tampilan Hasil Pencarian Dengan Keyword Salah Ketik

Seperti yang tertera pada Gambar 4.3 aplikasi menampilkan halaman hasil pencarian dengan kesalahan penulisan *keyword*. Halaman tersebut terdapat *search engine* dengan *keyword* yang telah diinputkan pada halaman *dashboard* atau dari halaman hasil pencarian itu sendiri. Kemudian terdapat keterangan waktu eksekusi yang dibutuhkan sistem untuk memeriksa kebenaran penulisan *keyword* termasuk dari mencocokkan dengan data berita pada *database* sampai dengan berhasil menampilkan data tersebut. Selanjutnya terdapat keterangan rekomendasi koreksi berdasarkan hasil perhitungan sistem.

Portal Berita Pariwisata Madura

Waktu eksekusi: 2.0

Tanggal	Judul
3/25/2022	Liburan Akhir Pekan di Pantai Ropet, Surga Pulau Gili Iyang yang Wajib Dikunjungi saat ke Sumenep
5/6/2022	Berlibur Lebaran Sambil Nikmati Live Musica di Pantai Ekasoghi Sumenep
5/12/2022	Pengunjung Pantai Lombang dan Slopeng Sumenep Membeludak, Ini PR Dinas Pariwisata
5/10/2022	Panduan Wisata ke Pantai Sembilan Sumenep, Termasuk Tarif Homestay
5/3/2022	Pantai Sembilan Kabupaten Sumenep Madura
17/1/2021	Nikmati Indahya Hamparan Pasir Putih Pantai Sembilan Sumenep

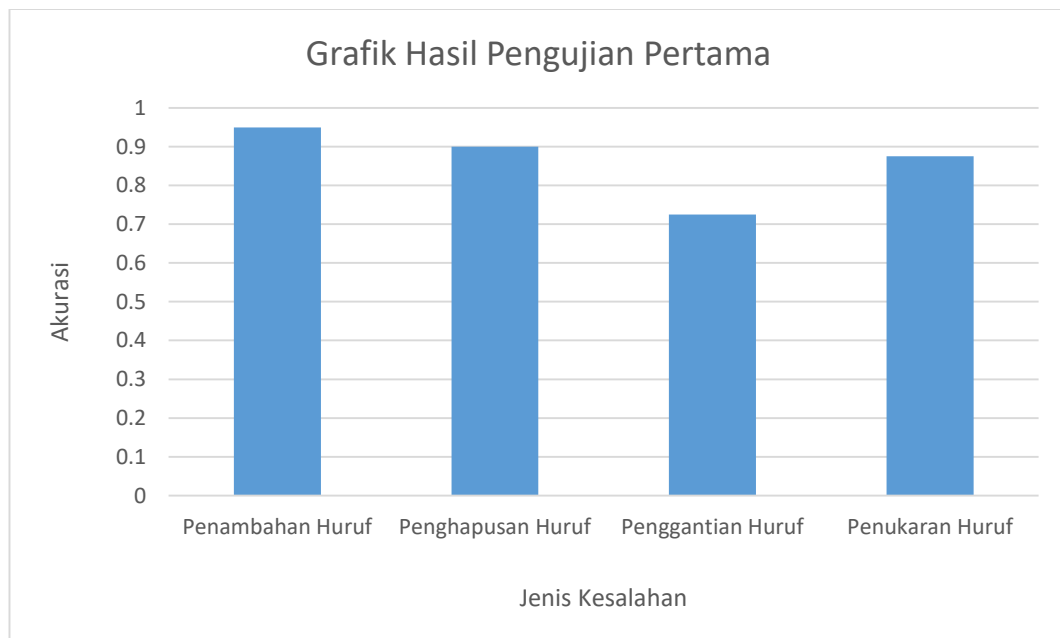
180411100135 - Muhammad Ali Yusuf

Gambar 4.4 Tampilan Hasil Pencarian Dengan Keyword Benar

Seperti pada Gambar 4.4 aplikasi menampilkan halaman hasil pencarian berdasarkan penulisan *keyword* yang benar atau tidak terdapat kesalahan penulisan. Pada halaman tersebut terdapat *search engine* dengan *keyword* yang telah diinputkan oleh *user* pada halaman *dashboard* atau dari halaman hasil pencarian itu sendiri. Kemudian terdapat keterangan waktu eksekusi yang dibutuhkan oleh sistem dalam memeriksa kebenaran penulisan *keyword* dengan mencocokkan kata yang terdapat pada kamus termasuk dari mencocokkan data berita dalam *database* sampai dengan aplikasi berhasil menampilkan data yang diinginkan *user*.

4.6 Evaluasi Sistem

Setelah sistem berhasil dibuat, langkah selanjutnya yakni evaluasi sistem yang menjelaskan kembali hasil pengujian dan menjabarkan kendala yang terjadi ataupun permasalahan lanjutan dari penelitian yang telah dilakukan. Berdasarkan hasil pengujian menunjukkan bahwa *Jaro-Winkler Distance* dapat melakukan perbaikan koreksi ejaan kata dengan baik pada beberapa kata yang kurang tepat atau yang terdapat kesalahan penulisan. Detail grafik hasil akurasi *Jaro-Winkler Distance* pada setiap jenis kesalahan kata dapat dilihat pada Gambar 4.5.

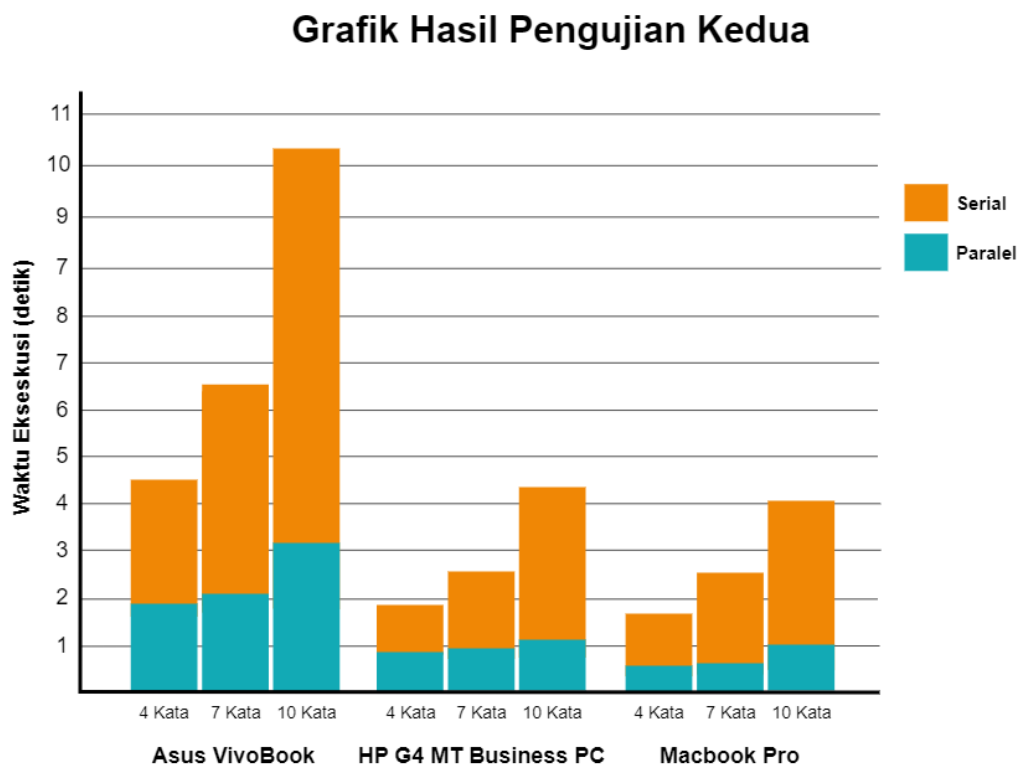


Gambar 4.5 Grafik Hasil Pengujian Pertama

Berdasarkan hasil pengujian pada setiap jenis kesalahan kata pada Gambar 4.5, didapatkan bahwa akurasi pada setiap skenario memiliki hasil yang berbeda-

beda. Nilai akurasi tertinggi terdapat pada jenis kesalahan penghapusan huruf dengan akurasi sebesar 0.95 dan akurasi terendah pada jenis kesalahan penggantian huruf dengan nilai akurasi 0.725. Sedangkan pada jenis kesalahan penambahan dan penukaran huruf masing-masing menghasilkan nilai akurasi sebesar 0.90 dan 0.875. Jika di rata-rata pada seluruh jenis kesalahan kata, algoritma *Jaro-Winkler Distance* menghasilkan akurasi dengan nilai 0.86. Adapun kendala atau permasalahan lanjutan dari penelitian ini yakni aplikasi yang dibuat masih belum mampu mengidentifikasi entitas-entitas seperti nama orang dan nama objek lainnya, sehingga dibutuhkan penambahan model untuk *Name Entity Recognition* (NER). Selain itu kata atau bahasa asing yang tidak terdapat pada *dataset* dan kata berulang seperti "tempat-tempat" masih dianggap salah oleh sistem.

Selanjutnya pada pengujian kedua menunjukkan bahwa model pemrosesan secara paralel menggunakan *multiprocessing* mampu mempercepat waktu eksekusi pada setiap perangkat yang diujikan. Detail hasil perbandingan waktu eksekusi yang didapatkan baik dengan model pemrosesan secara serial maupun paralel dapat dilihat pada Gambar 4.6.



Gambar 4.6 Grafik Hasil Pengujian Kedua

Dari hasil pengujian pemrosesan secara serial maupun paralel di 3 perangkat pada Gambar 4.6, didapatkan waktu eksekusi terpendek terdapat pada perangkat MacBook Pro baik pada proses secara serial maupun paralel dengan rasio perbandingan 1.76 : 0.61 detik pada jumlah kata uji sebanyak 4. Berdasarkan hasil pengujian kedua ini, dapat diambil kesimpulan bahwa terdapat beberapa faktor yang berpengaruh pada semakin lama atau tidaknya waktu eksekusi, yakni: jumlah kata input, jumlah ketersediaan CPU pada perangkat, performa CPU, dan banyaknya *dataset* juga mempengaruhi waktu pemrosesan. Sehingga *parallel processing* tidak menjamin waktu eksekusi dapat meningkat secara signifikan pada perangkat yang ketersediaan CPU yang tidak cukup banyak.

BAB 5

PENUTUP

Bab ini berisikan kesimpulan beserta saran dari penelitian yang telah dilakukan pada algoritma *Jaro-Winkler Distance* untuk Koreksi Ejaan Kata Menggunakan *Multiprocessing* pada Sistem Pencarian Berita Pariwisata, untuk dikembangkan pada penelitian selanjutnya.

5.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan, dapat disimpulkan bahwa algoritma *Jaro-Winkler Distance* dapat melakukan koreksi ejaan kata dengan baik. Hasil penelitian menunjukkan akurasi yang berbeda pada setiap jenis kesalahan penulisan. Nilai akurasi tertinggi terdapat pada jenis kesalahan penghapusan huruf dengan akurasi sebesar 0.95 dan akurasi terendah pada jenis kesalahan penggantian huruf dengan nilai akurasi 0.725. Pada jenis kesalahan penambahan dan penukaran huruf masing-masing menghasilkan nilai akurasi sebesar 0.90 dan 0.875. Sehingga jika di rata-rata dari semua jenis skenario kesalahan kata, maka nilai akurasi yang didapatkan sebesar 0.86.

Selanjutnya pengujian terhadap *parallel processing* dengan penerapan *multiprocessing* menunjukkan bahwa model pemrosesan secara paralel mampu mempercepat waktu eksekusi pada setiap perangkat yang diujikan. Waktu eksekusi terpendek terdapat pada perangkat MacBook Pro baik pada proses secara serial maupun paralel dengan rasio perbandingan 1.76 : 0.61 detik pada jumlah kata uji sebanyak 4. Namun di sisi lain waktu eksekusi bergantung dari jumlah *dataset* daftar kata yang digunakan. Semakin banyak data semakin lama waktu eksekusi dibutuhkan. *Parallel processing* juga tidak menjamin peningkatan waktu eksekusi, tergantung dari spesifikasi komputer yang digunakan terutama jumlah CPU pada perangkat.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, maka diperoleh saran sebagai berikut:

1. Diharapkan menggunakan kamus dengan versi terbaru yang lebih lengkap pada penelitian selanjutnya.
2. Pentingnya penambahan kamus entitas dalam penelitian berikutnya. Kata-kata entitas diperlukan untuk menyimpan atribut yang tidak terdapat pada kamus kata dasar seperti: nama orang, tempat, dan objek lainnya.
3. Pengembangan sistem dengan mengimplementasikan model *Named Entity Recognition* (NER) untuk pengecekan kata-kata entitas.
4. Penelitian ini dapat dikembangkan untuk mengganti kata yang salah secara otomatis dengan saran kata yang dihasilkan.

DAFTAR PUSTAKA

- [1] N. Fonna, *Pengembangan Revolusi Industri 4.0 dalam Berbagai Bidang*. 2019.
- [2] N. S. Bhirud, R. . Bhavsar, and B. . Pawar, “Grammar Checkers for Natural Languages,” *Int. J. Nat. Lang. Comput.*, vol. 6, no. 4, pp. 1–13, 2017, doi: 10.5121/ijnlc.2017.6401.
- [3] B. Leonardo and S. Hansun, “Text documents plagiarism detection using Rabin-Karp and Jaro-Winkler distance algorithms,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 5, no. 2, pp. 462–471, 2017, doi: 10.11591/ijeecs.v5.i2.pp462-471.
- [4] F. Breitingner, B. Guttman, M. McCarrin, and V. Roussev, “Approximate Matching: Definition and Terminology,” *Natl. Inst. Stand. Technol.*, 2014.
- [5] V. Sagita and M. I. Prasetyowati, “Studi Perbandingan Implementasi Algoritma Boyer-Moore, Turbo Boyer-Moore, dan Tuned Boyer-Moore dalam Pencarian String,” *J. Ultim.*, vol. 4, no. 1, pp. 31–37, 2013.
- [6] M. Syaroni and R. Munir, “Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching) Dalam Bahasa Inggris,” *Semin. Nas. Apl. Teknol. Inf. 2005*, pp. 1–6, 2005.
- [7] Y. Rochmawati and R. Kusumaningrum, “Studi Perbandingan Algoritma Pencarian String dalam Metode Approximate String Matching untuk Identifikasi Kesalahan Pengetikan Teks,” *J. Buana Inform.*, vol. 7, pp. 125–134, 2015.
- [8] I. B. K. S. Arnawa, “Implementasi Algoritma Levenshtein Pada Sistem Pencarian Judul Skripsi / Tugas Akhir,” *J. Sist. dan Inform.*, vol. 11, pp. 46–53, 2017.
- [9] Geovadi, “Indonesian Wordlist,” [Online]. Available: <https://github.com/geovedi/indonesian-wordlist/edit/master/01-kbbi3-2001-sort-alpha.lst>.

- [10] P. Norvig, "30.000 High Frequency English Vocabulary," [Online]. Available: <https://github.com/derekchuank/high-frequency-vocabulary>.
- [11] I. Cholissodin, R. Setya Perdana, and A. I. Fahma, "Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-gram dan Levenshtein Distance Optimasi Asupan Makanan Harian Ibu Hamil Penderita Hipertensi Menggunakan Algoritme Genetika View project Twitt," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 53–62, 2018.
- [12] A. Ririd, P. Y. Saputra, and A. M. Sastri, "Sistem Koreksi Kesalahan Pengetikan Kata Kunci dalam Pencarian Artikel Menggunakan Algoritma Jaro-Winkler," *Semin. Inform.*, pp. 60–65, 2019.
- [13] Rudi Haryanto, "Cerdas Jelajah Internet," *Depok: Kriya Pustaka*, 2009.
- [14] S. Yudho Baskoro, A. Ridok, and M. Tanzil Furqon, "Pencarian Pasal Pada Kitab Undang-Undang Hukum Pidana (Kuhp) Berdasarkan Kasus Menggunakan Metode Cosine Similarity Dan Latent Semantic Indexing (Lsi)," *J. Enviromental Eng. Sustain. Technol.*, vol. 2, no. 2, pp. 83–88, 2015, doi: 10.21776/ub.jeest.2015.002.02.4.
- [15] D. Abbott, "Introduction to Text Mining Virtual Data Intensive Summer School," *Behav. Insights Team*, 2013.
- [16] A. Kurniawati, S. Puspitodjati, and S. Rahman, "Implementasi Algoritma Jaro-Winkler Distance untuk Membandingkan Kesamaan Dokumen Berbahasa Indonesia," *Proceeding, Semin. Ilm. Nas. Komput. dan Sist. Intelijen KOMMIT 2008, Depok, Indones.*, 2010.
- [17] K. Manaf, S. W. Pitara, B. Subaeki, R. Gunawan, Rodiah, and Bakhtiar, "Comparison of Carp Rabin Algorithm and Jaro-Winkler Distance to Determine the Equality of Sunda Languages," *TSSA 2019 - 13th Int. Conf. Telecommun. Syst. Serv. Appl. Proc.*, pp. 77–81, 2019, doi: 10.1109/TSSA48701.2019.8985470.

- [18] D. Saunders, “What is Jaro-Winkler Similarity?,” *Baseclass*.
<https://www.baseclass.io/newsletter/jaro-winkler> (accessed Jan. 08, 2023).
- [19] R. Feldman and J. Sanger, *The Text Mining Handbook*. Boston: Cambridge University Press, 2007.
- [20] J. Palach, *Parallel Programming with Python*. Birmingham: Packt Publishing, 2014.
- [21] R. Gadde and J. Kelly, “Using Multiprocessing to Make Python Code Faster,” *Medium*, 2018. <https://urban-institute.medium.com/using-multiprocessing-to-make-python-code-faster-23ea5ef996ba> (accessed Jan. 31, 2023).
- [22] N. P. Lestari, “Uji Recall and Precision Sistem Temu Kembali,” *Univ. Airlangga*, vol. 5, no. 3, pp. 45–46, 2016.
- [23] J. Hasugian, “Penelusuran Informasi Ilmiah Secara Online: Perlakuan Terhadap Seorang Pencari Informasi Sebagai Real User,” *Pustaka*, vol. 2, no. 1, pp. 1-13–13, 2006.
- [24] A. Prasetyo, W. M. Baihaqi, and I. S. Had, “Algoritma Jaro-Winkler Distance: Fitur Autocorrect dan Spelling Suggestion pada Penulisan Naskah Bahasa Indonesia di BMS TV,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 4, p. 435, 2018, doi: 10.25126/jtiik.201854780.
- [25] O. R. Hartono, “Indonesian Stoplist,” 2017, [Online]. Available: <https://www.kaggle.com/datasets/oswinrh/indonesian-stoplist>.
- [26] Imamah and F. H. Rachman, “Twitter sentiment analysis of Covid-19 using term weighting TF-IDF and logistic regresion,” *Proceeding - 6th Inf. Technol. Int. Semin. ITIS 2020*, pp. 238–242, 2020, doi: 10.1109/ITIS50118.2020.9320958.
- [27] E. Setiani and W. Ce, “Text Classification Services Using Naïve Bayes for Bahasa Indonesia,” *Proc. 2018 Int. Conf. Inf. Manag. Technol. ICIMTech 2018*, pp. 361–366, 2018, doi: 10.1109/ICIMTech.2018.8528258.

LAMPIRAN

Lampiran 1. *Dataset* Berita Pariwisata

No.	url	tanggal	judul	kabupaten
1.	https://jatim.liputan6.com/read/4867530/pesona-bukit-jaddih-tebing-kapur-dengan-danau-biru-yang-cantik-di-madura	1/24/2022	Pesona Bukit Jaddih, Tebing Kapur dengan Danau Biru yang Cantik di Madura	Bangkalan
2.	https://www.tribunnewswiki.com/2022/02/03/pantai-siring-kemuning	2/3/2022	Pantai Siring Kemuning	Bangkalan
3.	https://jatim.liputan6.com/read/4981538/pesona-pantai-siring-kemuning-dengan-suasana-yang-indah-dan-memukau	6/8/2022	Pesona Pantai Siring Kemuning Dengan Suasana yang Indah dan Memukau	Bangkalan
4.	https://radarsolo.jawapos.com/wisata-kuliner/13/07/2022/dulu-tempat-sampah-kini-pantai-biru-bangkalan-jadi-destinasi-andalan/	7/13/2022	Dulu Tempat Sampah, Kini Pantai Biru Bangkalan Jadi Destinasi Andalan	Bangkalan
5.	https://tribunjatimtravel.tribunnews.com/2022/08/21/wisatawan-ramai-kunjungi-wisata-bukit-jaddih-di-bangkalan-madura	8/21/2022	Wisatawan Ramai Kunjungi Wisata Bukit Jaddih di Bangkalan Madura	Bangkalan
6.	https://surabaya.tribunnews.com/2022/08/21/tidak-ada-lagi-pungli-di-wisata-alam-bukit-jaddih-bangkalan-pengunjung-akui-makin-nyaman	8/21/2022	Tidak Ada Lagi Pungli di Wisata Alam Bukit Jaddih Bangkalan, Pengunjung Akui Makin Nyaman	Bangkalan
7.	https://madura.tribunnews.com/2022/09/11/peserta-lomba-renang-pelajar-di-wisata-alam-bukit-jaddih-bangkalan-batasi-kuota	9/11/2022	Peserta Lomba Renang Pelajar di Wisata Alam Bukit Jaddih Membeludak, Dispora Bangkalan Batasi Kuota	Bangkalan

No.	url	tanggal	judul	kabupaten
	jaddih-membeludak-dispora-bangkalan-batasi-kuota			
8.	https://www.detik.com/jatim/wisata/d-5942628/monumen-arek-lancor-di-pamekasan-tak-pernah-absen-dipadati-pengunjung	2/15/2022	Monumen Arek Lancor di Pamekasan Tak Pernah Absen Dipadati Pengunjung	Pamekasan
9.	https://surabaya.kompas.com/read/2022/04/06/054300478/sejarah-masjid-agung-assyuhada-pamekasan-simbol-perjuangan-mujahid-perang?page=all	4/6/2022	Sejarah Masjid Agung Assyuhada Pamekasan, Simbol Perjuangan Mujahid Perang	Pamekasan
10.	https://suryamalang.tribunnews.com/2022/05/03/pengunjung-wisata-di-pantai-talang-siring-dan-jumiang-meningkat-polres-pamekasan-gelar-operasi	5/3/2022	Pengunjung Wisata di Pantai Talang Siring dan Jumiang Meningkat, Polres Pamekasan Gelar Operasi	Pamekasan
11.	https://jatim.tribunnews.com/2022/05/07/pantai-talang-siring-pamekasan-jadi-tempat-favorit-berlibur-bareng-keluarga	5/7/2022	Pantai Talang Siring Pamekasan Jadi Tempat Favorit Berlibur Bareng Keluarga	Pamekasan
12.	https://www.tribunnewswiki.com/2022/05/14/pantai-jumiang	5/14/2022	Pantai Jumiang	Pamekasan
13.	https://www.tribunnewswiki.com/2022/05/16/pantai-talang-siring	5/16/2022	Pantai Talang Siring	Pamekasan
14.	https://maduraraya.jurnalisindonesia.id/read/99767/Pasarean-Joko-Tarub-Destinas-Wisata-Religi-di-Kabupaten-Pamekasan-yang-Sarat-Akan-Nilai-Sejarah	7/25/2022	Pasarean Joko Tarub, Destinasi Wisata Religi di Kabupaten Pamekasan yang Sarat Akan Nilai Sejarah	Pamekasan

No.	url	tanggal	judul	kabupaten
15.	https://www.indozone.id/travel/6gspO97/viral-api-abadi-pamekasan-bikin-warga-penasaran-suka-dipakai-masak-nasi-dan-bakar-jagung	8/15/2022	Viral Api Abadi Pamekasan Bikin Warga Penasaran, Suka Dipakai Masak Nasi dan Bakar Jagung	Pamekasan
16.	https://www.idntimes.com/travel/destination/ratna-herlina/info-wisata-pantai-jumiang-pamekasan-c1c2?page=all	9/12/2022	Info Wisata Pantai Jumiang Pamekasan: Lokasi, Rute, dan Aktivitas	Pamekasan
17.	https://www.indozone.id/travel/AqsAMeB/enggak-kalah-sama-kota-batu-pamekasan-punya-bukit-kehi-panoramanya-bikin-malas-pulang/read-all	9/19/2022	Enggak Kalah Sama Kota Batu! Pamekasan Punya Bukit Keki, Panoramanya Bikin Malas Pulang	Pamekasan
18.	https://jatim.tribunnews.com/2022/04/01/masjid-agung-sampang-mulai-siapkan-pelaksanaan-salat-tarawih	4/1/2022	Masjid Agung Sampang Mulai Siapkan Pelaksanaan Salat Tarawih	Sampang
19.	https://travel.detik.com/domestic-destination/d-6023311/ini-pantai-camplong-favorit-ngabuburit-anak-muda-sampang	4/8/2022	Ini Pantai Camplong, Favorit Ngabuburit Anak Muda Sampang	Sampang
20.	https://rri.co.id/sumenep/1141-religi/1424493galap-berkah-ramadhan-di-makam-aji-gunung	4/15/2022	Ngalap Berkah Ramadan di Makam Aji Gunung	Sampang
21.	https://surabaya.tribunnews.com/2022/05/05/pengunjung-wisata-pantai-camplong-sampang-membludak-personel-gabungan-beri-pengamanan-ekstra	5/5/2022	Pengunjung Wisata Pantai Camplong Sampang Membeludak, Personel Gabungan Beri Pengamanan Ekstra	Sampang

No.	url	tanggal	judul	kabupaten
22.	https://www.detik.com/jatim/wisata/d-6065493/pengunjung-pantai-camplong-sampang-membeludak-satpolair-dirikan-posko-sar	5/6/2022	Pengunjung Pantai Camplong Sampang Membeludak, Satpolair Dirikan Posko SAR	Sampang
23.	https://www.tribunnewswiki.com/2022/05/13/pantai-camplong	5/13/2022	Pantai Camplong	Sampang
24.	https://www.indozone.id/travel/YvsP5xR/air-terjun-terindah-di-pulau-garam-menyimpan-legenda-pasutri-yang-cekcok-hingga-mati/read-all	5/31/2022	Air Terjun Terindah di Pulau Garam Menyimpan Legenda Pasutri yang Cekkock hingga Mati	Sampang
25.	https://www.indozone.id/travel/Q8sd81X/pantainya-tak-berpasir-tapi-berbatu-karang-eksotis	1/5/2022	Pantainya Tak Berpasir, Tapi Berbatu Karang Eksotis	Sumenep
26.	https://www.tribunnewswiki.com/2022/02/03/pantai-slopeng	2/3/2022	Pantai Slopeng	Sumenep
27.	https://www.mongabay.co.id/2022/02/24/upaya-pulihkan-hutan-mangrove-pulau-kecil-di-sumenep/	2/24/2022	Upaya Pulihkan Hutan Mangrove Pulau Kecil di Sumenep	Sumenep
28.	https://www.detik.com/jatim/wisata/d-5983229/bakar-baronang-di-pantai-semilan-saat-langit-bertabur-bintang	3/15/2022	Bakar Baronang di Pantai Sembilan Saat Langit Bertabur Bintang	Sumenep
29.	https://tribunjatimtravel.tribunnews.com/2022/03/25/liburan-akhir-pekan-di-pantai-ropet-surga-pulau-gili-iyang-yang-wajib-dikunjungi-saat-ke-sumenep	3/25/2022	Liburan Akhir Pekan di Pantai Ropet, Surga Pulau Gili Iyang yang Wajib Dikunjungi saat ke Sumenep	Sumenep

No.	url	tanggal	judul	kabupaten
30.	https://jatim.tribunnews.com/2022/04/02/masjid-jamik-sumenep-siap-gelar-salat-tarawih-bakal-sediakan-takjil-gratis-selama-ramadan	4/2/2022	Masjid Jamik Sumenep Siap Gelar Salat Tarawih, Bakal Sediakan Takjil Gratis Selama Ramadan	Sumenep
31.	https://surabaya.kompas.com/read/2022/04/04/163000178/mengenal-gili-iyang-di-sumenep-pulau-dengan-kadar-oksigen-terbaik-nomor-2?page=all	4/4/2022	Mengenal Gili Iyang di Sumenep, Pulau dengan Kadar Oksigen Terbaik Nomor 2 di Dunia	Sumenep
32.	https://travel.detik.com/domestic-destination/d-6014960/sumenep-punya-tanah-lot-lho-tapi-tidak-ada-pura	4/4/2022	Sumenep Punya Tanah Lot Lho, tapi Tidak Ada Pura	Sumenep
33.	https://jatim.inews.id/berita/mengenal-gili-iyang-pulau-di-madura-dengan-kadar-oksigen-tertinggi-kedua-di-dunia	4/12/2022	Mengenal Gili Iyang, Pulau di Madura dengan Kadar Oksigen Tertinggi Kedua di Dunia	Sumenep
34.	https://rri.co.id/sumenep/feature/1422557/masjid-jamik-sumenep-masjid-tertua-dirancang-non-muslim-tionghoa?utm_source=news_main&utm_medium=internal_link&utm_campaign=General%20Campaign	4/13/2022	Masjid Jamik Sumenep, Masjid Tertua Dirancang Non Muslim Tionghoa	Sumenep
35.	https://sumenepkab.go.id/berita/baca/sejuta-keindahan-pantai-badur-siap-sambut-wisatawan	5/5/2022	Sejuta Keindahan Pantai Badur Siap Sambut Wisatawan	Sumenep
36.	https://beritajatim.com/gaya-hidup/berlibur-lebaran-sambil-nikmati-live-music-di-pantai-ekasoghi-sumenep/	5/6/2022	Berlibur Lebaran Sambil Nikmati Live Musica di Pantai Ekasoghi Sumenep	Sumenep

No.	url	tanggal	judul	kabupaten
37.	https://beritajatim.com/gaya-hidup/pengunjung-pantai-lombang-dan-slopeng-sumenep-membeludak-ini-pr-dinas-pariwisata/	5/12/2022	Pengunjung Pantai Lombang dan Slopeng Sumenep Membeludak, Ini PR Dinas Pariwisata	Sumenep
38.	https://jatim.liputan6.com/	6/13/2022	Pesona Pantai Lombang, Wisata Pantai Pasir Putih di Ujung Timur Madura	Sumenep
39.	https://asional.okezone.com/read/2022/07/04/337/2622937/menko-pmk-keraton-sumenep-berpotensi-jadi-wisata-nasional	7/4/2022	Menko PMK: Keraton Sumenep Berpotensi Jadi Wisata Nasional	Sumenep
40.	https://portalmadura.com/live-music-wisata-pantai-ekasoghi-manjakan-pengunjung-lebaran-iduladha-2022-292184/	7/9/2022	Live Music Wisata Pantai eKasoghi, Manjakan Pengunjung Lebaran Iduladha 2022	Sumenep
41.	https://jatim.tribunnews.com/	8/15/2022	Wisatawan Mulai Banyak Kunjungi Pulau Gili Labak Sumenep, Jatuh Hati Pada Aktivitas Snorkeling	Sumenep
42.	https://amp.kompas.com/	9/19/2022	Gili Labak : Daya Tarik, Aktivitas, dan Rute	Sumenep
43.	https://jatimnet.com/ini-wisata-terindah-di-sampang-wajib-dikunjungi-saat-di-madura	8/2/2020	Ini Wisata Terindah di Sampang, Wajib Dikunjungi Saat di Madura	Sampang
44.	https://www.liputan6.com/regional/read/3992836/riwayat-lon-malang-pantai-yang-sedang-hits-di-madura	6/20/2019	Riwayat Lon Malang, Pantai yang Sedang Hits di Madura	Sampang
45.	https://www.pulaumadura.com/2021/01/harga-tiket-fasilitas-dan-lokasi-maps-wisata-pantai-lon-malang-sampang.html.html	1/16/2021	Harga Tiket, Fasilitas dan Lokasi Maps Wisata Pantai Lon Malang - Sampang	Sampang
46.	https://radarsolo.jawapos.com/wisata-kuliner/29/06/2022/mengunjungi-pantai-lon-	6/29/2022	Mengunjungi Pantai Lon Malang, Sampang yang Dulu Dipenuhi Semak Semak	Sampang

No.	url	tanggal	judul	kabupaten
	malang-sampang-yang-dulu-dipenuhi-semak-semak/			
47.	https://travelspromo.com/htm-wisata/air-terjun-toroan-sampang/	9/2/2021	Wisata Air Terjun Toroan Yang Penuh Legenda	Sampang
48.	https://www.idntimes.com/travel/destination/ratna-herlina/air-terjun-toroan-clc2	12/30/2021	Air Terjun Toroan: Informasi Rute, Lokasi, dan Tips Berlibur	Sampang
49.	https://www.nativeindonesia.com/air-terjun-toroan/	11/24/2022	Air Terjun Toroan, Lokasi Wisata Air Terjun Di Tepi Pantai Nepa	Sampang
50.	https://www.inews.id/multimedia/photo/wisata-hutan-keranepa-di-sampang-yang-dihuni-500-monyet-ekor-panjang	7/2/2022	Wisata Hutan Kera Nepa di Sampang yang Dihuni 500 Monyet Ekor Panjang	Sampang
51.	https://www.tempatwisata.pro/wisata/Waduk-Klampis	19/9/2017	Waduk Klampis, Melihat Pesona Indahnya Salah Satu Waduk di Madura	Sampang
52.	https://www.kaskus.co.id/thread/60c47e105aaa8f500107a051/pesona-waduk-klampis-alternatif-wisata-di-madura/	12/6/2021	Pesona Waduk Klampis, Alternatif Wisata di Madura	Sampang
53.	https://www.pulaumadura.com/2016/03/obyek-wisata-alam-gua-lebar-di-kota.html?m=1	11/3/2016	Objek Wisata Alam Gua Lebar di Kota Sampang	Sampang
54.	https://surabaya.tribunnews.com/2018/10/30/madura-memiliki-gua-lebar-konon-jadi-tempat-pertapaan-pangeran-trunojoyo	30/10/2018	Madura Memiliki Gua Lebar, Konon Jadi Tempat Pertapaan Pangeran Trunojoyo	Sampang
55.	https://www.tempatwisata.pro/wisata/Pantai-Nepa	4/9/2017	Wisata Pantai Nepa, Menikmati Keindahan Pantai dan Hutan Keranya	Sampang
56.	https://jejakpiknik.com/pantai-nepa/	18/12/2022	Pantai Nepa Madura	Sampang

No.	url	tanggal	judul	kabupaten
57.	https://www.pulaumadura.com/2015/08/obyek-wisata-pantai-nepa-di-sampang.html	19/8/2015	Objek Wisata Pantai Nepa di Kabupaten Sampang	Sampang
58.	https://agelejer.com/daya-tarik-objek-wisata-bukit-masegit-yang-kian-mempesona/	14/7/2022	Daya Tarik Objek Wisata Bukit Masegit Yang Kian Mempesona	Sampang
59.	https://webinfowisata.wordpress.com/2016/08/11/tempat-wisata-baru-bukit-masegit-di-sampang-madura/	11/8/2016	Tempat Wisata Baru Bukit Masegit di Sampang Madura	Sampang
60.	https://www.kliktimes.com/travel/pr-7295442208/rekomendasi-destinasi-wisata-sampang-syahdu-dan-instagramable	3/11/2022	Rekomendasi Destinasi Wisata Sampang, Syahdu dan Instagramable	Sampang
61.	https://www.pegipegi.com/travel/arosbaya-bukit-kapur-cantik-mahakarya-manusia-di-madura/	5/11/2018	Arosbaya, Bukit Kapur Cantik Mahakarya Manusia di Madura	Bangkalan
62.	https://www.merdeka.com/travel/bukit-arosbaya-pahatan-relief-menakjubkan-di-tanah-madura.html	12/5/2021	Bukit Arosbaya, Pahatan Relief Menakjubkan di Tanah Madura	Bangkalan
63.	https://pojoksuramadu.com/bukit-kapur-arosbaya/	27/7/2022	Wisata Bukit Kapur Arosbaya Bangkalan Madura	Bangkalan
64.	https://panduanwisata.id/bukit-arosbaya-bangkalan/	29/7/2022	Bukit Arosbaya Bangkalan Madura Pahatan Batu Kapur Yang Menakjubkan	Bangkalan
65.	https://jurnalpost.com/wisata-baru-di-bancaran-bangkalan-telah-di-buka/17550/	29/1/2021	Wisata Baru di Bancaran Bangkalan Telah Di Buka	Bangkalan
66.	https://www.pulaumadura.com/2016/11/wisata-hutan-mangrove-bangkalan.html	8/11/2016	WOW! Ada Objek Wisata Alam Hutan Mangrove di Desa Labuhan - Bangkalan	Bangkalan

No.	url	tanggal	judul	kabupaten
67.	https://www.kompasiana.com/rijaltaufiqieramadlantzaman9138/6030f7fad541df43144a36c2/jelajah-tempat-eksotis-alami-di-hutan-mangrove-bangkalan-madura	20/2/2021	Jelajah Tempat Eksotis Alami di Hutan Mangrove Bangkalan, Madura	Bangkalan
68.	https://www.liputan6.com/regional/read/4643849/cerita-desa-labuhan-ekonomi-berseri-di-mangrove-yang-lestari	29/8/2021	Cerita Desa Labuhan: Ekonomi Berseri di Mangrove yang Lestari	Bangkalan
69.	https://petajatim.co/berburu-keindahan-sunset-pantai-tengket-nan-eksotis-surga-bagi-para-instagrammer/	7/11/2020	Berburu Keindahan Sunset Pantai Tengket Nan Eksotis, Surga Bagi Para Instagrammer	Bangkalan
70.	https://www.maduramu.com/image/detail/1587/pantai-tengket-dusun-maneron-kecamatan-sepulu-kabupaten-bangkalan	19/1/2022	Pantai Tengket, Dusun Maneron, Kecamatan Sepulu, Kabupaten Bangkalan	Bangkalan
71.	https://wesata.id/blog/pantai-tengket-wisata-bahari-tersembunyi-di-pulau-madura	16/7/2022	Pantai Tengket, Wisata Bahari Tersembunyi di Pulau Madura	Bangkalan
72.	https://jatim.poskota.co.id/2021/06/01/mengintip-pantai-tengket-yang-tersembunyi-di-kabupaten-bangkalan-dengan-panorama-indah	1/6/2021	Mengintip Pantai Tengket yang Tersembunyi di Kabupaten Bangkalan dengan Panorama Indah	Bangkalan
73.	https://madurapost.net/pantai-biru-bangkalan-destinasi-wisata-dari-bekas-pembuangan-sampah/	18/3/2021	Pantai Biru Bangkalan, Destinasi Wisata Dari Bekas Pembuangan Sampah	Bangkalan
74.	https://maduranewsmedia.com/terkini/pantai-biru-destinasi-wisata-baru-di-kabupaten-bangkalan-untuk-liburan-nataru/	24/12/2019	Pantai Biru Destinasi Wisata Baru Di Kabupaten Bangkalan Untuk Liburan Nataru	Bangkalan

No.	url	tanggal	judul	kabupaten
75.	https://www.pesonawisataindonesia.com/wisata-pantai-biru-madura/	27/7/2019	Wisata Pantai Biru Madura	Bangkalan
76.	https://pojoksuramadu.com/pantai-siring-kemuning/	31/8/2022	Rute Tiket Masuk Pantai Siring Kemuning Bangkalan Madura	Bangkalan
77.	https://kumparan.com/user-10042021085005/mencicipi-keindahan-pantai-siring-kemuning-di-madura-1vawMwuWqr0	21/4/2021	Mencicipi Keindahan Pantai Siring Kemuning di Madura	Bangkalan
78.	https://www.pulaumadura.com/2020/09/harga-tiket-fasilitas-dan-lokasi-maps-wisata-pantai-tlangoh-bangkalan.html	27/9/2020	Harga Tiket, Fasilitas dan Lokasi Maps Wisata Pantai Tlangoh - Bangkalan	Bangkalan
79.	https://pamekasankab.go.id/berita/73/objek-wisata-di-kabupaten-pamekasan	25/7/2017	Objek Wisata di Kabupaten Pamekasan	Pamekasan
80.	https://jejapiknik.com/bukit-cinta-pamekasan/	19/12/2022	Tiket Masuk Bukit Cinta Pamekasan Madura	Pamekasan
81.	https://www.sinergimadura.com/tentang-madura/pr-2904983532/wisata-bukit-cinta-pamekasan-madura-dan-11-fasilitasnya	1/10/2022	Wisata Bukit Cinta Pamekasan Madura dan 11 Fasilitasnya	Pamekasan
82.	https://kumparan.com/bangsaonline/bukit-cinta-destinasi-wisata-baru-di-pamekasan-yang-cocok-untuk-foto-foto	6/10/2017	Bukit Cinta, Destinasi Wisata Baru di Pamekasan yang Cocok untuk Foto foto	Pamekasan
83.	https://www.madureh.com/2019/11/bukit-cinta-pamekasan-madura.html	3/11/2019	Bukit Cinta Pamekasan Madura : Harga Tiket dan Lokasi	Pamekasan
84.	https://portalwisata.co.id/bukit-brukoh-wisata-di-pamekasan-madura/	28/7/2019	Bukit Brukoh, Wisata di Pamekasan Madura yang Populer dan Hits 2020	Pamekasan

No.	url	tanggal	judul	kabupaten
85.	https://tempatwisataunik.com/wisata-indonesia/jawa-timur/tempat-wisata-di-pamekasan/attachment/bukit-brukoh-pakong	17/3/2020	Bukit Brukoh Pakong	Pamekasan
86.	https://www.suarakarya.id/metro/pr-2601649295/Wisata-Bukit-Brukoh-Pamekasan-Surga-Lokasi-Selfie-Bagi-Wisatawan	3/10/2018	Wisata Bukit Brukoh Pamekasan, Surga Lokasi Selfie Bagi Wisatawan	Pamekasan
87.	https://lingkarmadura.pikiran-rakyat.com/pariwisata-budaya/pr-1892952108/vihara-avalokitesvara-pamekasan-vihara-unik-dengan-pura-hindu-dan-langgar-di-dalamnya	5/11/2021	Vihara Avalokitesvara Pamekasan, Vihara Unik dengan Pura Hindu dan Langgar di Dalamnya	Pamekasan
88.	https://timesindonesia.co.id/peristiwa-daerah/374986/vihara-avalokitesvara-tempat-ibadah-umat-tri-darma-terbesar-di-madura	9/10/2021	Vihara Avalokitesvara Tempat Ibadah Umat Tri Darma Terbesar di Madura	Pamekasan
89.	https://agelejer.com/wisata-air-terjun-bulangan/	12/8/2022	Wisata Air Terjun Bulangan	Pamekasan
90.	https://alvilaaveniaaa.wordpress.com/2019/01/25/3-air-terjun-cantik-di-madura/	25/1/2019	3 Air Terjun Cantik di Pamekasan, Madura	Pamekasan
91.	https://www.sinergimadura.com/tentang-madura/pr-2904982866/tak-hanya-satu-di-pamekasan-terdapat-3-wisata-air-terjun-yang-bisa-kamu-kunjungi?page=2	1/10/2022	Tak Hanya Satu, Di Pamekasan Terdapat 3 Wisata Air Terjun yang Bisa Kamu Kunjungi!	Pamekasan
92.	https://www.unews.id/pendidikan/pr-2884801938/tempat-wisata-pantai-batu-kerbuy-dan-gua-blaban-di-pamekasan-yuk-mampir	19/9/2022	Tempat Wisata Pantai Batu Kerbuy dan Gua Blaban di Pamekasan, Yuk Mampir	Pamekasan

No.	url	tanggal	judul	kabupaten
93.	https://www.unews.id/travel/pr-2884980284/wisata-tebing-waru-pamekasan-suguhkan-pemandangan-alam-besar-dan-keasrian-bumi-gerbang-salam	1/10/2022	Wisata Tebing Waru Pamekasan, Suguhkan Pemandangan Alam dan Keasrian Bumi Gerbang Salam	Pamekasan
94.	https://travel.kompas.com/read/2022/10/05/210400927/panduan-wisata-ke-pantai-sembilan-sumenep-termasuk-tarif-homestay?page=all	5/10/2022	Panduan Wisata ke Pantai Sembilan Sumenep, Termasuk Tarif Homestay	Sumenep
95.	https://pojoksuramadu.com/indahnyapantai-sembilan-di-kabupaten-sumenep-madura/	5/3/2022	Pantai Sembilan Kabupaten Sumenep Madura	Sumenep
96.	https://www.jatimpos.co/pariwisata/4290-nikmati-indahnya-hamparan-pasir-putih-pantai-sembilan-sumenep	17/1/2021	Nikmati Indahnya Hamparan Pasir Putih Pantai Sembilan Sumenep	Sumenep
97.	https://kumparan.com/jendela-dunia/harga-tiket-masuk-bukit-tinggi-daramista-wisata-hits-di-jawa-timur-1yL54exBsIJ	25/5/2022	Harga Tiket Masuk Bukit Tinggi Daramista, Wisata Hits di Jawa Timur	Sumenep
98.	https://sinarjateng.pikiran-rakyat.com/nasional/pr-1006017086/destinasi-wisata-alam-bukit-tinggi-daramista-di-kabupaten-sumenep-jawa-timur-yang-terkenal-keindahannya	26/12/2022	Destinasi Wisata Alam Bukit Tinggi Daramista di Kabupaten Sumenep Jawa Timur yang Terkenal Keindahannya	Sumenep
99.	https://wisatakita.com/wisata/Jawa.Timur/Sumenep/Museum.Keraton.Sumenep	19/10/2018	Museum Keraton Sumenep	Sumenep
100.	https://agelejer.com/pesona-air-terjun-duko-yang-tak-tertandingi/	28/7/2022	Pesona Air Terjun Duko yang Tak Tertandingi	Sumenep

Lampiran 2. *User Relevance Judgement*

Hasil Pengujian Skenario Penambahan Huruf							
No.	Kata Salah	Kata Benar	Hasil Sistem	Validator 1	Validator 2	Validator 3	Hasil Validasi
1.	wisatta	wisata	wisata	1	1	1	1
2.	panatai	pantai	panai	0	0	0	0
3.	inndah	indah	indah	1	1	1	1
4.	hutana	hutan	hutan	1	1	1	1
5.	bukkit	bukit	bukit	1	1	1	1
6.	pullau	pulau	pulau	1	1	1	1
7.	gunnung	gunung	gunung	1	1	1	1
8.	panaorama	panorama	panorama	1	1	1	1
9.	tikket	tiket	tiket	1	1	1	1
10.	suneset	sunset	sunset	1	1	1	1
11.	rekomenedasi	rekomendasi	rekomendasi	1	1	1	1
12.	waduuk	waduk	waduk	1	1	1	1
13.	kuotta	kuota	kuota	1	1	1	1
14.	bnuka	buka	buka	1	1	1	1
15.	possko	posko	posko	1	1	1	1
16.	liburran	liburan	liburan	1	1	1	1
17.	monummen	monumen	monumen	1	1	1	1
18.	musieum	museum	museum	1	1	1	1

19.	peSSona	peSona	peSona	1	1	1	1
20.	populler	populer	populer	1	1	1	1
21.	tebbing	tebing	tebing	1	1	1	1
22.	suassana	suasana	suasana	1	1	1	1
23.	desitinasi	destinasi	desis	0	0	0	0
24.	allam	alam	alam	1	1	1	1
25.	religih	religi	religi	1	1	1	1
26.	kabhupaten	kabupaten	kabupaten	1	1	1	1
27.	virral	viral	viral	1	1	1	1
28.	rutte	rute	rute	1	1	1	1
29.	lokkasi	lokasi	lokasi	1	1	1	1
30.	informassi	informasi	informasi	1	1	1	1
31.	favorrit	favorit	favorit	1	1	1	1
32.	legghenda	legenda	legenda	1	1	1	1
33.	ekksotis	eksotis	eksotis	1	1	1	1
34.	manggrove	mangrove	mangrove	1	1	1	1
35.	dunnia	dunia	dunia	1	1	1	1
36.	nassional	nasional	nasional	1	1	1	1
37.	kerraton	keraton	keraton	1	1	1	1
38.	hargha	harga	harga	1	1	1	1
39.	fasillitas	fasilitas	fasilitas	1	1	1	1
40.	massuk	masuk	masuk	1	1	1	1
Akurasi							0.95

Hasil Pengujian Skenario Penghapusan Huruf							
No.	Kata Salah	Kata Benar	Hasil Sistem	Validator 1	Validator 2	Validator 3	Hasil Validasi
1.	wsata	wisata	wisata	1	1	1	1
2.	pntai	pantai	pantai	1	1	1	1
3.	inda	indah	indah	1	1	1	1
4.	utan	hutan	utang	0	0	0	0
5.	buki	bukit	bukit	1	1	1	1
6.	pulu	pulau	puluh	0	0	0	0
7.	gunug	gunung	gunung	1	1	1	1
8.	panrama	panorama	panorama	1	1	1	1
9.	tiet	tiket	tiket	1	1	1	1
10.	snset	sunset	sunset	1	1	1	1
11.	rekomedasi	rekomendasi	rekomendasi	1	1	1	1
12.	wduk	waduk	waduk	1	1	1	1
13.	kuot	kuota	kuota	1	1	1	1
14.	uka	buka	ukas	0	0	0	0
15.	posk	posko	posko	1	1	1	1
16.	libuan	liburan	liburan	1	1	1	1
17.	mnumen	monumen	monumen	1	1	1	1
18.	museu	museum	museum	1	1	1	1
19.	psona	pesona	pesona	1	1	1	1
20.	pouler	populer	populer	1	1	1	1

21.	tbing	tebing	tebing	1	1	1	1
22.	suasna	suasana	suasana	1	1	1	1
23.	detinasi	destinasi	destinasi	1	1	1	1
24.	alm	alam	alam	1	1	1	1
25.	reigi	religi	religi	1	1	1	1
26.	kaupaten	kabupaten	kabupaten	1	1	1	1
27.	virl	viral	viral	1	1	1	1
28.	rte	rute	rote	0	0	0	0
29.	lokai	lokasi	lokasi	1	1	1	1
30.	iformasi	informasi	informasi	1	1	1	1
31.	favori	favorit	favorit	1	1	1	1
32.	legeda	legenda	legenda	1	1	1	1
33.	esotis	eksotis	eksotis	1	1	1	1
34.	mangrov	mangrove	mangrove	1	1	1	1
35.	dnia	dunia	dunia	1	1	1	1
36.	nasinal	nasional	nasional	1	1	1	1
37.	kerton	keraton	keraton	1	1	1	1
38.	harg	harga	harga	1	1	1	1
39.	fasilita	fasilitas	fasilitas	1	1	1	1
40.	msuk	masuk	masuk	1	1	1	1
Akurasi							0.9

Hasil Pengujian Skenario Penggantian Huruf							
No.	Kata Salah	Kata Benar	Hasil Sistem	Validator 1	Validator 2	Validator 3	Hasil Validasi
1.	wisats	wisata	wisata	1	1	1	1
2.	pantao	pantai	pamit	0	0	0	0
3.	indsh	indah	indah	1	1	1	1
4.	hutam	hutan	hutan	1	1	1	1
5.	bikit	bukit	biskuit	0	0	0	0
6.	oulau	pulau	pulau	1	1	1	1
7.	gunung	gunung	gunung	1	1	1	1
8.	pamorama	panorama	panorama	1	1	1	1
9.	tijet	tiket	tiket	1	1	1	1
10.	sumset	sunset	sunset	1	1	1	1
11.	rekomensasi	rekomendasi	rekomendasi	1	1	1	1
12.	wadik	waduk	waduk	1	1	1	1
13.	kiota	kuota	kita	0	0	0	0
14.	buja	buka	bujal	0	0	0	0
15.	oosko	posko	otoskop	0	0	0	0
16.	libiran	liburan	liburan	1	1	1	1
17.	mpnumen	monumen	monumen	1	1	1	1
18.	museun	museum	museum	1	1	1	1
19.	oesona	pesona	pesona	1	1	1	1
20.	popiler	populer	populer	1	1	1	1

21.	tebing	tebing	tebing	1	1	1	1
22.	auasana	suasana	alasan	0	0	0	0
23.	desrinasi	destinasi	destinasi	1	1	1	1
24.	alqm	alam	alam	1	1	1	1
25.	rekigi	religi	religi	1	1	1	1
26.	ksbupsten	kabupaten	kabupaten	1	1	1	1
27.	virak	viral	irak	0	0	0	0
28.	rite	rute	ritme	0	0	0	0
29.	lpkasi	lokasi	lokasi	1	1	1	1
30.	imformasi	informasi	informasi	1	1	1	1
31.	faforit	favorit	favorit	1	1	1	1
32.	leyenda	legenda	legenda	1	1	1	1
33.	exsotis	eksotis	eksotis	1	1	1	1
34.	mangrofe	mangrove	mangrove	1	1	1	1
35.	dumia	dunia	dumi	0	0	0	0
36.	nasionsl	nasional	nasional	1	1	1	1
37.	keraron	keraton	keran	0	0	0	0
38.	harha	harga	hara	0	0	0	0
39.	fasikitas	fasilitas	fasilitas	1	1	1	1
40.	mssuk	masuk	masuk	1	1	1	1
Akurasi							0.725

Hasil Pengujian Skenario Penukaran Huruf							
No.	Kata Salah	Kata Benar	Hasil Sistem	Validator 1	Validator 2	Validator 3	Hasil Validasi
1.	wiasta	wisata	wisata	1	1	1	1
2.	pantia	pantai	pantai	1	1	1	1
3.	indha	indah	indah	1	1	1	1
4.	huatn	hutan	hutan	1	1	1	1
5.	bkuit	bukit	bukit	1	1	1	1
6.	pulua	pulau	pula	0	0	0	0
7.	gunugn	gunung	gunung	1	1	1	1
8.	panoraam	panorama	panorama	1	1	1	1
9.	itket	tiket	tiket	1	1	1	1
10.	sunste	sunset	sunset	1	1	1	1
11.	reokmenadsi	rekomendasi	rekomendasi	1	1	1	1
12.	wadku	waduk	waduk	1	1	1	1
13.	kouta	kuota	kota	0	0	0	0
14.	buak	buka	buka	1	1	1	1
15.	opsko	posko	posko	1	1	1	1
16.	liubran	liburan	liburan	1	1	1	1
17.	monumne	monumen	monumen	1	1	1	1
18.	musuem	museum	museum	1	1	1	1
19.	peosna	pesona	pesona	1	1	1	1
20.	populre	populer	populer	1	1	1	1

21.	teibng	tebing	tebing	1	1	1	1
22.	sausana	suasana	sasana	0	0	0	0
23.	destiansi	destinasi	destinasi	1	1	1	1
24.	aalm	alam	alam	1	1	1	1
25.	reilgi	religi	religi	1	1	1	1
26.	kbaupatne	kabupaten	kabupaten	1	1	1	1
27.	ivral	viral	viral	1	1	1	1
28.	rtue	rute	rute	1	1	1	1
29.	loaksi	lokasi	lokasi	1	1	1	1
30.	inforamsi	informasi	informasi	1	1	1	1
31.	favoirt	favorit	favorit	1	1	1	1
32.	legedna	legenda	legenda	1	1	1	1
33.	ekostis	eksotis	eksotis	1	1	1	1
34.	mangroev	mangrove	mangrove	1	1	1	1
35.	dunai	dunia	dunia	1	1	1	1
36.	nasoinla	nasional	nasional	1	1	1	1
37.	kearton	keraton	keraton	1	1	1	1
38.	hagra	harga	hara	0	0	0	0
39.	faislitas	fasilitas	fasilitas	1	1	1	1
40.	amsuk	masuk	amuk	0	0	0	0
Akurasi							0.875

DAFTAR RIWAYAT HIDUP



Data Pribadi:

Nama : Muhammad Ali Yusuf

Jenis Kelamin : Laki-Laki

Tempat, Tanggal Lahir : Bangkalan, 22 September 2001

Alamat : Perumahan Cendana 1 Blok C / No. 07

Kelurahan/Desa : Banyuajuh

Kecamatan : Kamal

Kabupaten : Bangkalan

Agama : Islam

Status Perkawinan : Belum Menikah

Kewarganegaraan : Indonesia

Prodi : Teknik Informatika

No. Hp : 087862660616

E-mail : muhammadaliyusuff22@gmail.com

No.	Pendidikan Formal	Tahun
1.	MIN 1 Bangkalan	2008-2014
2.	MTs Amanatul Ummah Surabaya	2014-2016
3.	MA Amanatul Ummah Surabaya	2016-2018
4.	Universitas Trunojoyo Madura	2018-2023