

**DETEKSI KESALAHAN EJA KATA LULUH PADA BERITA
DENGAN ALGORITMA JACCARD SIMILARITY
(STUDI KASUS : TRIBUNNEWS)**



SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh
Gelar Sarjana Komputer (S.Kom.)

**Nicholas Evan
00000027900**

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG**

2023

**DETEKSI KESALAHAN EJA KATA LULUH PADA BERITA
DENGAN ALGORITMA JACCARD SIMILARITY
(STUDI KASUS : TRIBUNNEWS)**



Diajukan sebagai salah satu syarat untuk memperoleh
Gelar Sarjana Komputer (S.Kom.)

**Nicholas Evan
00000027900**

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA
PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG
2023

HALAMAN PERNYATAAN TIDAK PLAGIAT

Dengan ini saya,

Nama : Nicholas Evan

Nomor Induk Mahasiswa : 00000027900

Program Studi : Informatika

Skripsi dengan judul:

Deteksi Kesalahan Eja Kata Luluh pada Berita dengan Algoritma Jaccard Similarity (Studi Kasus : Tribunnews)

merupakan hasil karya saya sendiri bukan plagiat dari karya ilmiah yang ditulis oleh orang lain, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar serta dicantumkan di Daftar Pustaka.

Jika di kemudian hari terbukti ditemukan kecurangan/ penyimpangan, baik dalam pelaksanaan Skripsi maupun dalam penulisan laporan Skripsi, saya bersedia menerima konsekuensi dinyatakan TIDAK LULUS untuk Tugas akhir yang telah saya tempuh.

Tangerang, 3 Januari 2023



(Nicholas Evan)

UMM
UNIVERSITAS
MULTIMEDIA
NUSANTARA

HALAMAN PENGESAHAN

Skripsi dengan judul

**DETEKSI KESALAHAN EJA KATA LULUH PADA BERITA
DENGAN ALGORITMA JACCARD SIMILARITY
(STUDI KASUS : TRIBUNNEWS)**

oleh

Nama : Nicholas Evan
NIM : 00000027900
Program Studi : Informatika
Fakultas : Fakultas Teknik dan Informatika

Telah diujikan pada hari Senin, 9 Januari 2023

Pukul 10.00 s/s 11.30 dan dinyatakan

LULUS

Dengan susunan penguji sebagai berikut

Ketua Sidang



(Moeljono Widjaja, B.Sc., M.Sc., Ph.D.)

NIDN: 0311106903

Penguji



(Angga Aditya Permana, S.Kom.,
M.Kom.)

NIDN: 0407128901

Pembimbing

(Marlinda Vasty Overbeek, S.Kom, M.Kom)

NIDN: 0818038501

Ketua Program Studi Informatika,

(Marlinda Vasty Overbeek, S.Kom., M.Kom.)

NIDN: 0818038501

HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Multimedia Nusantara, saya yang bertanda tangan di bawah ini:

Nama : Nicholas Evan
NIM : 00000027900
Program Studi : Informatika
Fakultas : Teknik dan Informatika
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada **Universitas Multimedia Nusantara** hak Bebas Royalti Non-eksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

DETEKSI KESALAHAN EJA KATA LULUH PADA BERITA DENGAN ALGORITMA JACCARD SIMILARITY (STUDI KASUS : TRIBUNNEWS)

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non eksklusif ini Universitas Multimedia Nusantara berhak menyimpan, mengalih media / format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis / pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Tangerang, 3 Januari 2023

Yang menyatakan

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Nicholas Evan

Halaman Persembahan / Motto

"When you get tired, learn to rest, not to quit."

Banksy



KATA PENGANTAR

Puji Syukur atas berkat dan rahmat kepada Tuhan Yang Maha Esa, atas selesainya penulisan laporan Skripsi ini dengan judul: Deteksi Kesalahan Eja Kata Luluh pada Berita dengan Algoritma Jaccard Similarity (Studi Kasus : Tribunnews) dilakukan untuk memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer Jurusan Informatika Pada Fakultas Teknik dan Informatika Universitas Multimedia Nusantara. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Bapak Dr. Ninok Leksono, selaku Rektor Universitas Multimedia Nusantara.
2. Dr. Eng. Niki Prastomo, S.T., M.Sc., selaku Dekan Fakultas Teknik dan Informatika Universitas Multimedia Nusantara.
3. Ibu Marlinda Vasty Overbeek, S.Kom., M.Kom., selaku Ketua Program Studi Informatika Universitas Multimedia Nusantara dan Pembimbing pertama yang telah banyak meluangkan waktu untuk memberikan bimbingan, arahan dan motivasi atas terselesainya skripsi ini.
4. Keluarga dan teman-teman yang telah memberikan bantuan dukungan material dan moral, sehingga penulis dapat menyelesaikan tugas akhir ini.
5. Jessica Augustine S. yang telah memberikan dukungan dan bantuan moral, selama pengerjaan tugas akhir ini.
6. Jerico Olwen, selaku teman seperjuangan selama pengerjaan tugas akhir.

Semoga skripsi ini bermanfaat, baik sebagai sumber informasi maupun sumber inspirasi, bagi para pembaca.

Tangerang, 3 Januari 2023



Nicholas Evan

**DETEKSI KESALAHAN EJA KATA LULUH PADA BERITA
DENGAN ALGORITMA JACCARD SIMILARITY
(STUDI KASUS : TRIBUNNEWS)**

Nicholas Evan

ABSTRAK

Bahasa Indonesia merupakan bahasa nasional yang digunakan dalam kehidupan sehari-hari, namun kesalahan berbahasa kerap terjadi dalam di sekitar kita, salah satunya pada portal berita *online*. Kesalahan berbahasa merupakan penyimpangan bahasa dari kaidah tata bahasa dan salah satunya adalah peluluhan fonem. Hal ini terjadi akibat penulisan dilakukan secara *manual* sehingga memungkinkan untuk terjadinya kesalahan pengetikan. Dengan terjadinya kesalahan peluluhan ini, dilakukanlah penelitian yaitu pembuatan sistem dengan menggunakan algoritma *Jaccard Similarity* untuk mendeteksi kesalahan eja pada kata terluluh. *Jaccard Similarity* merupakan algoritma yang digunakan untuk membandingkan dokumen untuk menghitung kesamaan nilai dari dua dokumen. Evaluasi dilakukan dengan menggunakan *confusion matrix* yang kemudian diambil *F-1 score*-nya selain itu efisiensi sistem juga diperhitungkan. Hasil deteksinya memiliki *F-1 score* sebesar 66.6% dan efisiensi sistem dipengaruhi oleh jumlah kalimat dan jumlah kata yang terluluh. Sistem yang dibangun dapat mendeteksi kesalahan eja pada kata terluluh saat dihadapkan dengan berita dari portal berita Tribun.

Kata kunci: berita, *Jaccard Similarity*, kesalahan eja, peluluhan fonem, sistem deteksi



Detection of Spelling Errors in Indonesian Language News with Jaccard Similarity Algorithm (Case : Tribun News)

Nicholas Evan

ABSTRACT

Language is an organized communication tool in the form of units such as words, groups of words, clauses and sentences. Indonesian is the national language which should be used in accordance with Enhanced Indonesian Spelling, but language errors often occur accidentally on online news portals. Language errors are language deviations from grammatical rules and one of them is phoneme decay. This may occur due to writing done manually so that it allows typing errors. With the occurrence of this error, research was carried out, namely making a system using the Jaccard Similarity algorithm to detect misspellings in melted words. Jaccard Similarity is an algorithm used to compare documents to calculate the similarity of values from two documents. Evaluation is done by using the confusion matrix which then takes the F-1 score besides that system efficiency is also taken into account. The detection results have an F-1 score of 66.6% and system efficiency is influenced by the number of sentences and the number of words decayed. The system built can detect misspelled words when dealing with news from the Tribun news portal.

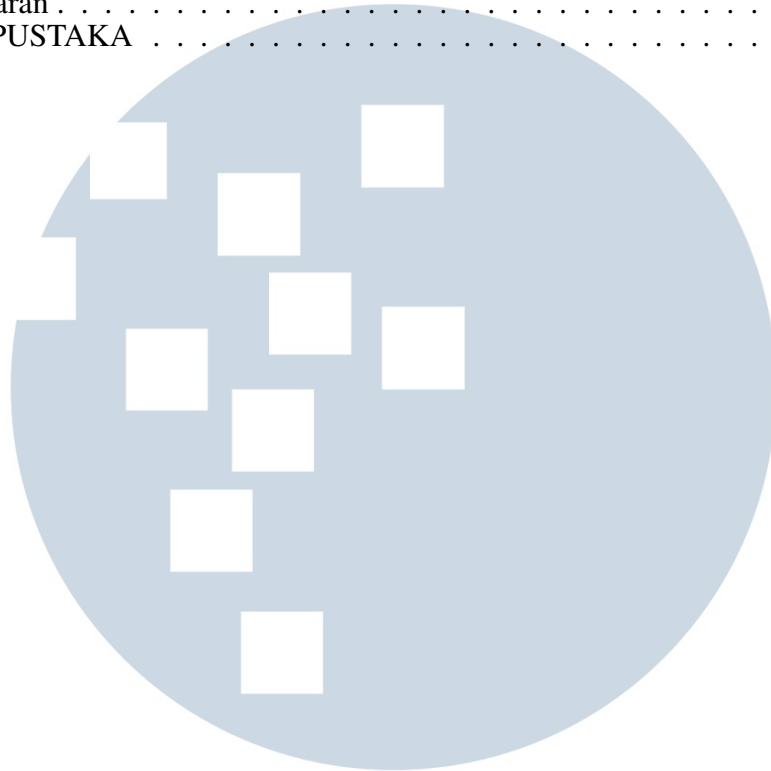
Keywords: *detection system, Jaccard Similarity, misspelling, news, phoneme decay*



DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN TIDAK MELAKUKAN PLAGIAT	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSETUJUAN PUBLIKASI ILMIAH	iv
HALAMAN PERSEMBAHAN/MOTO	v
KATA PENGANTAR	vi
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
DAFTAR KODE	xiii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Permasalahan	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	3
BAB 2 LANDASAN TEORI	5
2.1 Tinjauan Teori	5
2.1.1 Portal Berita	5
2.1.2 TRIBUN NEWS	5
2.1.3 Natural Language Processing	6
2.1.4 Text Preprocessing	6
2.1.5 Jaccard Similarity	7
2.1.6 Confusion Matrix	7
BAB 3 METODOLOGI PENELITIAN	10
3.1 Pengumpulan Data	10
3.2 Proses Data menjadi Dataset	10
3.3 Praproses	11
3.4 Jaccard Similarity	12
BAB 4 HASIL DAN DISKUSI	14
4.1 Spesifikasi Sistem	14
4.2 Implementasi	14
4.2.1 Pengumpulan Data	14
4.2.2 Dataset	15
4.2.3 Praproses	17
4.2.4 Jaccard Similarity	19
4.3 Uji Coba	20
4.3.1 Uji Coba Benar	20
4.3.2 Uji Coba Kesalahan	22
4.3.3 Uji Coba Dengan Parameter	23
4.3.4 Perhitungan Confusion Matrix	28
4.4 Evaluasi	28
4.4.1 Evaluasi Confusion Matrix	28
4.4.2 Evaluasi Efisiensi Sistem	29

BAB 5	SIMPULAN DAN SARAN	31
5.1	Simpulan	31
5.2	Saran	31
DAFTAR PUSTAKA	32



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

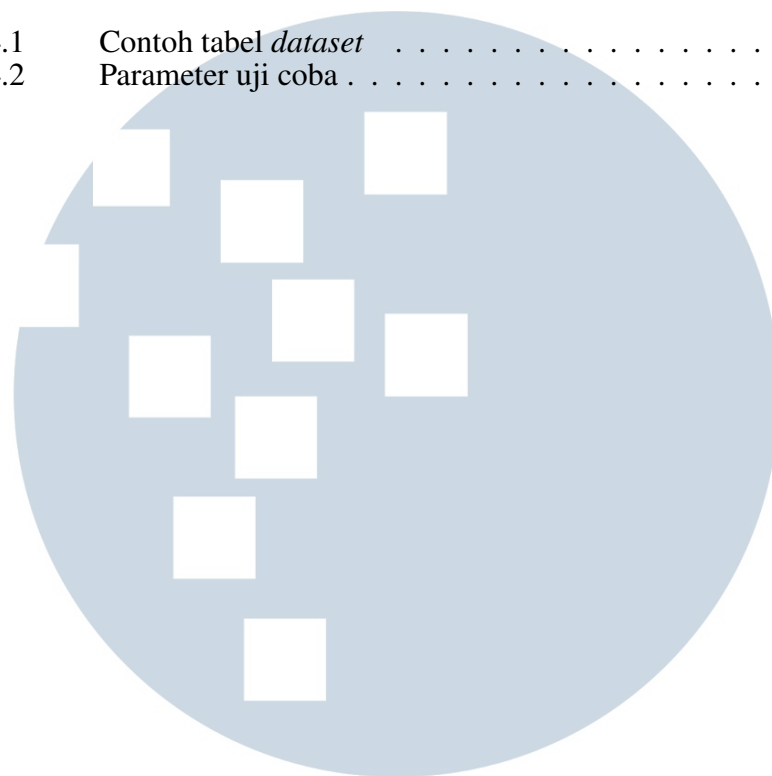
DAFTAR GAMBAR

Gambar 2.1	Logo Tribunnews	5
Gambar 2.2	Confusion matrix	8
Gambar 3.1	Diagram alir metodologi penelitian	10
Gambar 3.2	Diagram alir praproses	11
Gambar 3.3	Diagram alir perhitungan <i>jaccard similarity</i>	13
Gambar 4.1	Contoh data berita	15
Gambar 4.2	Contoh data dalam excel	15
Gambar 4.3	Potongan berita benar	21
Gambar 4.4	Hasil uji coba potongan berita benar	22
Gambar 4.5	Potongan berita salah	22
Gambar 4.6	Hasil uji coba potongan berita salah	23
Gambar 4.7	Hasil uji coba parameter 2 kalimat 25 kata	24
Gambar 4.8	Hasil uji coba parameter 2 kalimat 44 kata	25
Gambar 4.9	Hasil uji coba parameter 4 kalimat 55 kata	26
Gambar 4.10	Hasil uji coba parameter 4 kalimat 95 kata	26
Gambar 4.11	Hasil uji coba parameter 6 kalimat 124 kata	27
Gambar 4.12	Hasil uji coba parameter 6 kalimat 148 kata	27
Gambar 4.13	Confusion matrix hasil uji coba	28
Gambar 4.14	Evaluasi uji coba efisiensi	30
Gambar 4.15	Grafik uji coba efisiensi	30



DAFTAR TABEL

Tabel 4.1	Contoh tabel <i>dataset</i>	16
Tabel 4.2	Parameter uji coba	23



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

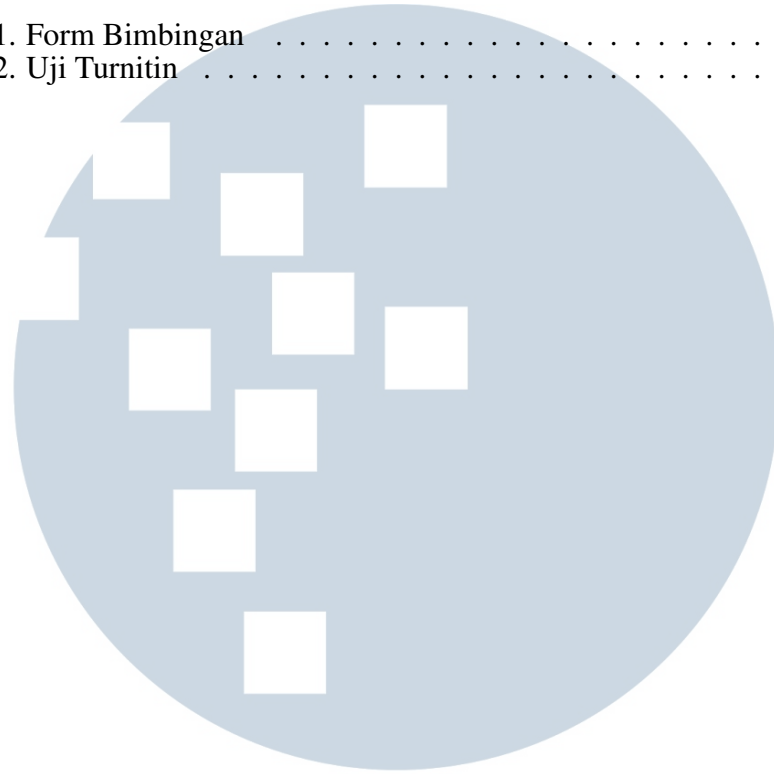
DAFTAR KODE

4.1	Potongan kode membaca file excel	16
4.2	Potongan kode membuat dataset	17
4.3	Potongan kode melakukan filtering	17
4.4	Potongan kode import regular expression	18
4.5	Potongan kode import string	18
4.6	Potongan kode case folding	18
4.7	Potongan kode tokenisasi kata	18
4.8	Potongan kode formula jaccard similarity	19
4.9	Potongan kode pengecekan dan perhitungan jaccard similarity	19



DAFTAR LAMPIRAN

Lampiran 1. Form Bimbingan	33
Lampiran 2. Uji Turnitin	35



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

BAB 1 PENDAHULUAN

1.1 Latar Belakang Masalah

Bahasa adalah alat komunikasi yang terorganisasi dalam bentuk satuan-satuan, seperti kata, kelompok kata, klausa dan kalimat yang diungkapkan baik secara lisan maupun tulis [1]. Bahasa Indonesia merupakan bahasa nasional yang dapat digunakan untuk berkomunikasi dengan berbagai suku yang ada di Indonesia, namun di generasi sekarang ini, bahasa yang seharusnya dipelajari dan dilestarikan malah disepelekan karena menganggap dirinya sudah bisa berbahasa Indonesia. Bahasa Indonesia yang seharusnya digunakan sesuai Ejaan Bahasa Indonesia yang Disempurnakan dan kaidah-kaidah kebahasaan malah terpengaruh oleh bahasa asing sehingga Ejaan Yang Disempurnakan dan kaidah-kaidah kebahasaan tidak lagi diperdulikan. Permasalahan lain juga ditemukan pada pembuatan kalimat, pengejaan kata, dan penggunaan tanda baca yang tidak sesuai.

Kesalahan berbahasa kerap terjadi pada portal berita *online*, menurut Jasmani melalui penelitiannya menyimpulkan bahwa masih banyak terdapat kesalahan-kesalahan berbahasa dari segi ejaan [2]. Hal ini terjadi akibat penulisan dilakukan secara *manual* sehingga memungkinkan untuk terjadinya kesalahan pengetikan. Menurut Djuraid, berita adalah suatu laporan ataupun pemberitahuan mengenai terjadinya peristiwa atau keadaan bersifat umum dan baru saja terjadi, yang disampaikan oleh wartawan media massa [3]. Membaca berita juga memiliki berbagai macam manfaat seperti mengetahui informasi yang aktual dan faktual, mengetahui kondisi aktual di tempat lain, menambah wawasan pembaca dan mengetahui serta memahami penulisan yang baik dan benar. Namun berita dapat memiliki kesalahan penulisan seperti penggunaan tanda baca dan huruf yang tidak sesuai, kesalahan penggunaan konjungsi, kesalahan penulisan kata, kesalahan ketik, dan penggunaan bahasa gaul, padahal salah satu manfaat dari membaca berita adalah untuk memahami cara penulisan yang baik dan benar.

Kesalahan berbahasa merupakan penyimpangan bahasa dari kaidah tata bahasa atau dari faktor-faktor cara berkomunikasi dan berbahasa lainnya yang telah ditentukan atau telah ditentukan dengan sendirinya [4]. Kesalahan ini dapat terjadi secara tidak sengaja, keliru, maupun memang tidak sesuai dengan tata bahasa yang bersangkutan. Salah satu kesalahan yang kerap terjadi adalah peluluhan fonem.

Peluluhan fonem merupakan bagian dari morfofonemik berarti perubahan fonem yang terjadi sebagai akibat pertemuan antara morfem (kata atau suku kata) yang satu dan morfem lain [5]. Tujuan dari adanya peluluhan fonem adalah mempermudah pelafalan dan menurut Penyuluh Kebahasaan dari Badan Pengembangan dan Pembinaan Bahasa Kemendikbud Wisnu Sasangka, peluluhan fonem pada kata dasar berawalan huruf k,p,s dan t terjadi karena adanya kemufakatan para pakar dan penutur bahasa [5]. Dengan melakukan pendeteksian kesalahan ketik terhadap kata terluluh tentunya akan membantu mengatasi kesalahan penulisan sehingga berita dapat diterbitkan dengan baik dan benar.

Dengan adanya permasalahan kesalahan pengetikan pada kata terluluh, dilakukanlah penelitian ini untuk mendeteksi kesalahan penulisan dalam berita berbahasa Indonesia. Pendeteksian kesalahan pengetikan kata terluluh dibangun dengan menggunakan algoritma *Jaccard Similarity*. Metode *jaccard similarity* dilakukan dengan menghitung kemiripan dari dua buah data, semakin mirip maka hasil angka yang dikeluarkan akan mendekati angka 1 dan sebaliknya. Hasil menunjukkan bahwa *Jaccard coefficient* dapat bekerja dengan baik dalam menghitung kemiripan dari dua buah data ketika membandingkan setiap huruf dalam kata[6]. Oleh karena itu, hasil penelitian ini diharapkan dapat mengurangi kesalahan ketik dalam peluluhan kata dan mempermudah pengecekan berita terhadap kesalahan penulisan sehingga berita dapat diterbitkan dengan menggunakan bahasa yang sesuai dan tepat.

1.2 Rumusan Masalah

Masalah yang dirumuskan dalam penelitian ini adalah sebagai berikut:

1. Bagaimana cara mendeteksi kesalahan pengetikan kata terluluh pada berita berbahasa Indonesia menggunakan algoritma *Jaccard Similarity*?
2. Bagaimana tingkat akurasi model yang telah dibuat dalam mendeteksi kesalahan pengetikan kata terluluh pada berita berbahasa Indonesia?

1.3 Batasan Permasalahan

Batasan-batasan yang digunakan dalam penelitian ini adalah sebagai berikut:

1. *Dataset* yang digunakan berasal dari berita TRIBUNNEWS.

2. Model yang dibuat hanya mendeteksi kalimat yang memiliki kata terluluh dengan awalan S,P,T, dan K.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disebutkan, terbentuklah tujuan dari penelitian ini, yaitu:

1. Mengimplementasikan algoritma *Jaccard Similarity* untuk pendeteksian kesalahan pengetikan kata terluluh pada berita berbahasa Indonesia.
2. Mengetahui tingkat akurasi model yang telah dibuat dalam mendeteksi kesalahan pengetikan kata terluluh pada berita berbahasa Indonesia.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat membantu para penulis berita dalam mengurangi kesalahan pengetikan kata terluluh dalam berita berbahasa Indonesia, dengan menggunakan sistem yang telah dibuat dengan menerapkan algoritma *Jaccard Similarity*.

1.6 Sistematika Penulisan

Berisikan uraian singkat mengenai struktur isi penulisan laporan penelitian, dimulai dari Pendahuluan hingga Simpulan dan Saran.

Sistematika penulisan laporan adalah sebagai berikut:

- Bab 1 PENDAHULUAN
Bab ini terdiri dari latar belakang masalah, rumusan-rumusan masalah, batasan permasalahan, tujuan penelitian, manfaat penelitian yang diharapkan, dan sistematika penulisan.
- Bab 2 LANDASAN TEORI
Bab ini berisikan tinjauan teori, yaitu teori-teori atau pengertian tidak umum yang digunakan untuk menjelaskan kepada pembaca agar dapat mengerti lebih dalam.
- Bab 3 METODOLOGI PENELITIAN
Bab ini menjelaskan metodologi yang dilakukan selama penelitian berlangsung.

- Bab 4 HASIL DAN DISKUSI

Bab ini menjelaskan implementasi, uji coba dan hasil evaluasi dari uji coba yang telah dilakukan.

- Bab 5 KESIMPULAN DAN SARAN

Bab ini menyatakan kesimpulan berdasarkan hasil evaluasi yang telah dilakukan selama penelitian.



BAB 2

LANDASAN TEORI

2.1 Tinjauan Teori

Tinjauan teori merupakan sebuah landasan yang digunakan dalam penelitian, berikut adalah tinjauan teori yang digunakan :

2.1.1 Portal Berita

Portal berita adalah media komunikasi online untuk pengguna internet membaca berita di seluruh dunia. Pengembangan portal berita memungkinkan untuk merilis publikasi, *press releases*, artikel, blog, dan konten yang berhubungan dengan berita. Dengan kata lain sebuah portal berita merupakan sebuah jalur akses terhadap berita[7].

2.1.2 TRIBUN NEWS



Gambar 2.1. Logo Tribunnews

Sumber : <https://logos.fandom.com/wiki/Tribunnews.com>

Tribunnews.com merupakan situs media online nomor satu di Indonesia. TribunNews memiliki media jaringan yang tersebar di penjuru Indonesia. Berpusat di Jakarta, Tribunnews.com merupakan media akselerasi transformasi digital Indonesia, hadir untuk menyajikan informasi dari seluruh penjuru Indonesia dari Sabang hingga Merauke melalui jaringan Tribun Network.

Sebagai media online terdepan Indonesia, Tribunnews.com diperkuat dengan *tagline* Mata Lokal Menjangkau Indonesia. *Hyperlocal* adalah misi Tribunnews.com berakar dari keyakinan bahwa setiap dari kita adalah orang lokal yang perlu terus melestarikan nilai dan perspektif setiap daerah ke seluruh Indonesia[8].

2.1.3 Natural Language Processing

Natural language processing adalah bagian dari Artificial Intelligence (AI) yang memberikan komputer kemampuan untuk membaca, mengerti dan mengartikan teks dan kata yang diucapkan dengan cara yang sama seperti yang dilakukan manusia[9]. Bahasa manusia sulit dimengerti oleh komputer karena data yang tidak terstruktur, selain itu kurangnya peraturan formal terhadap bahasa menyebabkan banyak teknik yang harus digunakan dalam NLP untuk menerjemahkannya kepada komputer. Tugas NLP adalah untuk memecah teks dan data suara menjadi hal yang masuk akal bagi komputer[10].

2.1.4 Text Preprocessing

Text preprocessing adalah suatu proses untuk menyeleksi data text agar menjadi lebih terstruktur lagi dengan melalui serangkaian tahapan yang meliputi tahapan *case folding*, *tokenizing*, *filtering*, dan *stemming*. *Text preprocessing* merupakan salah satu implementasi dari *text mining*. *Text mining* sendiri adalah suatu kegiatan menambang data, dimana data yang biasanya diambil berupa text yang bersumber dari dokumen-dokumen yang memiliki goals untuk mencari kata kunci yang mewakili dari sekumpulan dokumen tersebut sehingga nantinya dapat dilakukan analisa hubungan antara dokumen-dokumen tersebut[11]. Berikut adalah tahapan-tahapan *text preprocessing* menurut Nugroho[12].

1. *Case Folding*, salah satu bentuk *text preprocessing* yang paling sederhana dan efektif. Tujuan dari *case folding* adalah untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. *case folding* dapat berupa mengubah text menjadi huruf kecil, menghapus angka, dan juga menghapus karakter kosong.
2. *Tokenizing*, sebuah proses pemisahan teks menjadi potongan-potongan yang disebut sebagai *token* untuk kemudian di analisa. *Tokenizing* dapat dibagi lagi menjadi dua yaitu *tokenizing* kata dan *tokenizing* kalimat, dimana *tokenizing* kalimat digunakan untuk memisahkan kalimat dari paragraf sedangkan *tokenizing* kata memisahkan kata dari sebuah kalimat.
3. *Filtering*, sebuah tahap mengambil kata-kata penting dari hasil token dengan menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting).

4. *Stemming*, proses mengubah kata ke bentuk dasarnya,

Tujuan dari *text preprocessing* adalah untuk mewakili setiap dokumen sebagai vektor yaitu untuk membagi teks menjadi kata-kata individual. Dokumen teks dibentuk sebagai transaksi. Memilih kata kunci melalui proses pemilihan fitur dan langkah utama *text preprocessing* diperlukan untuk pengindeksan dokumen. Di sisi lain, tahap *text preprocessing* setelah membaca *input* dokumen teks adalah menjalankan fitur yang telah ditentukan[13].

2.1.5 Jaccard Similarity

Jaccard Similarity adalah sebuah algoritma yang memiliki fungsi untuk membandingkan dokumen dan menghitung kesamaan nilai dari dua objek atau dokumen[14]. Dokumen harus memiliki relasi / kesamaan sehingga dapat dibandingkan dengan dokumen lainnya. Formula dari Jaccard Similarity adalah irisan (*intersection*) dua dokumen dibagi dengan gabungan (*union*) dua dokumen. Berikut adalah formula dari Jaccard Similarity.

$$Sim_{jaccard} = \frac{|A \cap B|}{|A \cup B|} \quad (2.1)$$

Dengan keterangan sebagai berikut:

A = Dokumen 1

B = Dokumen 2

2.1.6 Confusion Matrix

Confusion Matrix adalah sebuah tabel yang sering digunakan untuk mengukur kinerja dari model klasifikasi di *machine learning*[15]. Mengukur kinerja suatu model yang telah dibuat merupakan langkah penting dalam *machine learning* sehingga dapat menjadi pertimbangan untuk memilih model terbaik[12]. Terdapat empat istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Gambar di bawah merupakan tabel *confusion matrix*.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Confusion Matrix

Gambar 2.2. Confusion matrix

Sumber : Nugroho, 2020

Keterangan dari Gambar 2.2 adalah :

1. *True Positive* (TP), merupakan data positif yang diprediksi benar
2. *True Negative* (TN), merupakan data negatif yang diprediksi benar
3. *False Positive* (FP), merupakan data negatif namun diprediksi sebagai data positif
4. *False Negative* (FN), merupakan data positif namun diprediksi sebagai data negatif.

Confusion matrix dapat digunakan untuk menghitung berbagai *performance metrics* untuk mengukur kinerja model yang telah dibuat. *Performance metrics* yang sering digunakan adalah : *accuracy*, *precision*, dan *recall*.

1. *Accuracy* menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. *accuracy* merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Nilai *accuracy* bisa didapatkan dengan formula sebagai berikut :

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.2)$$

2. *Precision* menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. *precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Nilai *precision* bisa didapatkan dengan menggunakan formula sebagai berikut :

$$precision = \frac{TP}{TP + FP} \quad (2.3)$$

3. *Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. *recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Nilai *recall* bisa didapatkan dengan menggunakan formula sebagai berikut :

$$recall = \frac{TP}{TP + FN} \quad (2.4)$$

4. *F-1 Score* menggambarkan perbandingan rata-rata precision dan recall yang dibobotkan[16]. Nilai *F-1* bisa didapatkan dengan menggunakan formula sebagai berikut :

$$F - 1Score = \frac{2 * recall * precision}{recall + precision} \quad (2.5)$$

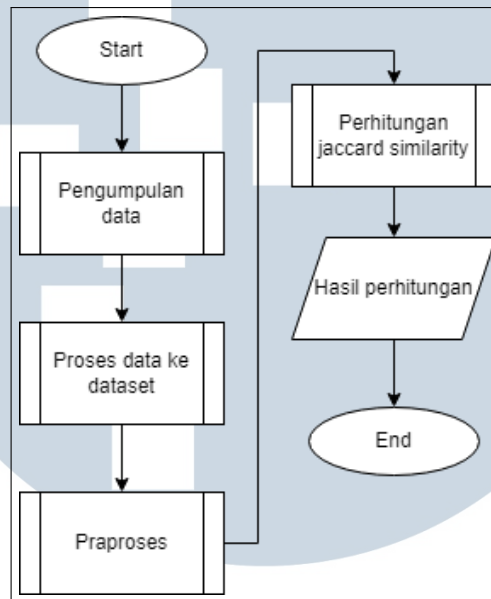
a



BAB 3

METODOLOGI PENELITIAN

Metode penelitian yang dilakukan untuk melaksanakan penelitian dari awal hingga akhir penelitian adalah sebagai berikut :



Gambar 3.1. Diagram alir metodologi penelitian

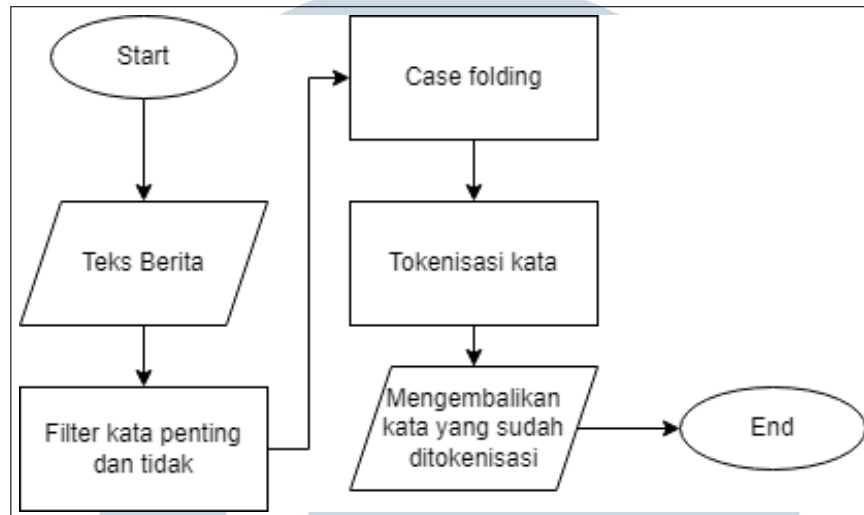
3.1 Pengumpulan Data

Pada tahap ini, data yang digunakan untuk penelitian merupakan berita Tribun News yang mengandung kesalahan pada pengejaannya dan telah diterbitkan. Data yang diberikan mengandung beberapa kesalahan seperti salah ketik, kata tidak baku, dan kesalahan-kesalahan lainnya. Data didapatkan dari kurang lebih 100 hingga 150 berita dengan hasil kata terluluhnya berjumlah 252 kata.

3.2 Proses Data menjadi Dataset

Pada tahap ini, data yang sudah didapatkan kemudian dicrawl kembali untuk mendapatkan kata terluluh baik yang benar maupun salah penyetikannya. Kata terluluh yang salah kemudian dibenarkan secara manual. Kemudian kata-kata terluluh yang sudah dikumpulkan dijadikan dataset untuk digunakan sebagai kamus kata terluluh yang benar.

3.3 Praproses



Gambar 3.2. Diagram alir praproses

Tahap praproses memiliki beberapa langkah yang akan dilakukan. Berikut adalah langkah-langkahnya :

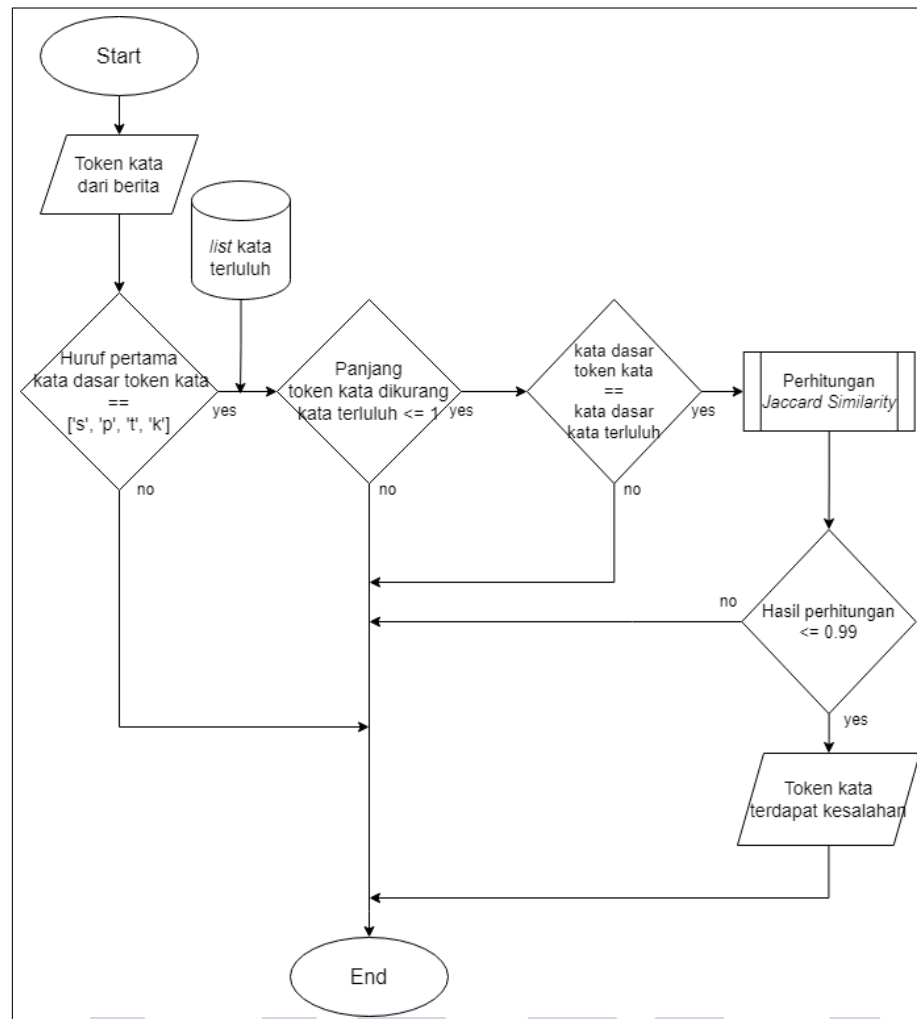
1. Langkah awal yang dilakukan adalah melakukan *filter* terhadap kata-kata yang terdapat di dalam berita atau artikel. *Filter* terjadi pada kata-kata yang dianggap tidak penting dengan menggunakan algoritma *stoplist*. Kata-kata dianggap tidak penting yaitu kata yang frekuensinya tinggi dan tidak memiliki makna, sehingga proses hanya dilakukan pada kata yang memiliki makna atau penting.
2. *Case folding*, merupakan salah satu bentuk teks preproses yang bertujuan untuk menyetarakan teks dengan mem-proses teks, yang pada kasus ini adalah artikel atau berita. *Case folding* memiliki beberapa langkah yaitu :
 - (a) Mengubah teks menjadi huruf kecil, tujuan mengubah berita atau artikel menjadi huruf kecil adalah mengantisipasi terjadinya kekeliruan dalam perhitungan menggunakan formula *Jaccard Similarity*.
 - (b) Menghapus angka, hal ini dilakukan karena angka merupakan hal yang tidak relevan dalam penelitian, angka bukan merupakan kata dan angka tidak akan dihitung dalam formula. Sehingga angka sebaiknya dihapus agar tidak diproses.

- (c) Menghapus tanda baca, hal ini juga dilakukan karena tanda baca tidak relevan dan tidak akan digunakan dalam perhitungan. Sehingga tanda baca sebaiknya dihapus agar tidak diproses.
 - (d) Menghapus *whitespace* atau karakter kosong, hal ini digunakan untuk menghapus spasi berlebih di awal dan akhir kata atau kalimat.
3. Tokenisasi kata, tokenisasi adalah pemotongan kata dari teks yang kemudian disebut sebagai token. Pada penelitian ini, token hanya berupa kata yang telah diubah menjadi huruf kecil tanpa ada tanda baca dan angka.
 4. Mengembalikan kata yang sudah ditokenisasi, proses ini mengembalikan *list of words* atau daftar kata dari artikel atau berita yang telah diproses sebelumnya.

3.4 Jaccard Similarity

Data yang telah dipraproses akan dihitung dengan formula jaccard similarity yang kemudian akan ditentukan hasilnya apakah kata terluluh sesuai dengan kata yang terdapat di dataset atau tidak.





Gambar 3.3. Diagram alir perhitungan *jaccard similarity*

Gambar 3.3 merupakan diagram alir perhitungan *jaccard similarity*. Fungsi ini menerima masukan berupa token-token kata dari berita, kemudian token kata tersebut diubah menjadi kata dasar yang kemudian dicek apakah huruf pertama dari kata tersebut adalah s / p / t / k, jika ya maka proses akan dilanjutkan dengan memeriksa apakah jumlah huruf yang dimiliki kata oleh berita dengan kata dari *dataset* jaraknya adalah 1, jika jaraknya adalah 1 atau kurang dari 1 maka proses akan dilanjutkan dengan melakukan pengecekan kata dasar dari token kata terhadap kata dari *dataset* apakah sama atau tidak. Kata-kata yang dapat melewati proses pengecekan akan dihitung similaritasnya dengan kata yang terdapat di dataset, jika hasil perhitungan similaritasnya di bawah 0.99 maka token kata dianggap salah atau tidak sesuai peluluhan.

BAB 4

HASIL DAN DISKUSI

4.1 Spesifikasi Sistem

Spesifikasi yang digunakan dalam pengerjaan penelitian ini adalah sebagai berikut :

1. Perangkat Keras

- Processor : AMD Ryzen 7 6800H 3.2Ghz (16 CPU)
- RAM : 16 GB 4800 MHz
- Graphics : NVIDIA GeForce RTX 3050Ti
- Storage : SSD 477GB

2. Perangkat Lunak

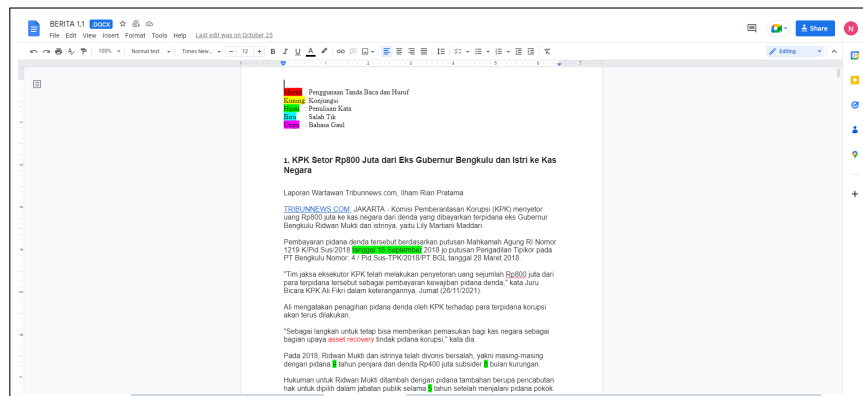
- Windows 11 64-bit
- Google Chrome
- Google Colaboration

4.2 Implementasi

Pada penelitian ini, dilakukan beberapa langkah implementasi seperti pengumpulan data, mengubah data menjadi dataset untuk digunakan sebagai kamus, melakukan langkah praproses, dan menjalankan perhitungan *jaccard similarity*.

4.2.1 Pengumpulan Data

Pengumpulan data dilakukan dengan dicatat secara manual pada excel terhadap berita yang diterima dari pihak Tribun. Data yang diterima berupa *file* google docs yang berisikan kumpulan-kumpulan berita.



Gambar 4.1. Contoh data berita

Gambar 4.1 adalah contoh berita yang diberikan oleh pihak Tribun. Berita-berita yang telah diberikan kemudian di-*crawl* untuk mendapatkan kata-kata terluluh yang digunakan dalam berita. Jumlah kata terluluh yang terkumpul dari kurang lebih 100 hingga 150 berita adalah 252 kata terluluh.

kata benar
menyamarkan
mengatakan
menyebabkan
menyentuh
mengonfirmasi
penangkapan
menuntut
menyebut
menangkap
menerangkan
menutup
penyampaian
menyindir
menirukan
menerapkan
menyampaikan
penanda
menandakan
memastikan
menambahkan

Gambar 4.2. Contoh data dalam excel

Gambar 4.2 menunjukkan contoh data yang telah dimasukkan ke dalam excel.

4.2.2 Dataset

Data yang telah dimiliki dalam excel kemudian diimpor ke dalam Google Colab yang selanjutnya akan digunakan untuk menjadi kamus kata terluluh yang benar.

```

1 import pandas as pd
2 dataset = pd.read_excel('kata_yang_benar.xlsx')
3

```

Kode 4.1: Potongan kode membaca file excel

Potongan Kode 4.1 adalah langkah yang dilakukan untuk membaca *file* excel dan kemudian diubah ke dalam bentuk tabel seperti pada Tabel 4.1.

Tabel 4.1. Contoh tabel *dataset*

kata benar
menyamarkan
mengatakan
menyebabkan
menyentuh
mengonfirmasi
penangkapan
menuntut
menyebut
menangkap
menerangkan
menutup
penyampaian
menyindir
menirukan
menerapkan
menyampaikan
penanda
menandakan
memastikan
menambahkan

Setelah data diubah menjadi tabel, hal selanjutnya yang harus dilakukan adalah mengubahnya menjadi *array* atau *list* untuk mempermudah dalam melakukan pengecekan.

```
1 data = dataset[ 'kata benar' ].to_numpy()
```

Kode 4.2: Potongan kode membuat dataset

Pada potongan Kode 4.2, variabel data digunakan untuk mengubah tabel dataset menjadi *array*.

4.2.3 Praproses

Praproses merupakan suatu tahapan yang dilakukan untuk menghilangkan atau mengubah text yang tidak digunakan selama proses eksekusi kode. Pada tahap praproses, hal-hal yang dilakukan adalah *stopwords / filtering*, *case folding*, menghilangkan *whitespace*, menghilangkan tanda baca, dan tokenisasi kata.

A *Stopwords / Filtering*

Filtering digunakan untuk membuang kata-kata yang tidak diperlukan atau tidak akan dicek seperti kata konjungsi dan sebagainya, hal ini dilakukan untuk mempercepat performa karena mengurangi jumlah kata yang akan masuk ke pengecekan. Langkah *Filtering* dilakukan dengan menggunakan *library* dari Sastrawi seperti potongan Kode 4.3.

```
1 !pip install sastrawi
2 from Sastrawi.StopWordRemover.StopWordRemoverFactory import
  StopWordRemoverFactory
3
4 factory_stopwords = StopWordRemoverFactory()
5 stopword = factory_stopwords.create_stop_word_remover()
6 stopword_remover = stopword.remove(news_html_free)
```

Kode 4.3: Potongan kode melakukan filtering

Langkah ini memfilter variabel `news_html_free` yang berisikan artikel berita yang diinput.

B *Case Folding*

Pada *case folding* terdapat beberapa langkah yang dilakukan, seperti :

1. Mengubah semua huruf menjadi huruf kecil, hal ini dilakukan untuk menyamaratakan huruf dalam artikel. Proses ini dilakukan dengan menggunakan metode yang terdapat di Python yaitu fungsi `lower()`.

2. Menghilangkan angka, karena dalam kasus penelitian ini angka tidak akan digunakan dalam pengecekan. Proses ini dilakukan dengan pertama mengimpor re atau regular expression.

```
1 import re
2
```

Kode 4.4: Potongan kode import regular expression

3. Menghilangkan *whitespace* atau *space* berlebih, proses ini dilakukan dengan menggunakan metode Python yaitu fungsi `strip()`.
4. Menghilangkan tanda baca untuk membantu dalam proses tokenisasi. Proses ini memerlukan impor tambahan yaitu,

```
1 import string
2
```

Kode 4.5: Potongan kode import string

Hal ini dilakukan untuk mencari tanda baca dan dihilangkan.

```
1 #mengubah huruf menjadi huruf kecil
2 lower_case = stopword_remover.lower()
3
4 #menghilangkan angka
5 numbering_removed = re.sub(r"\d+", "", lower_case)
6
7 #menghilangkan kelebihan whitespace/karakter kosong
8 whitespace_removed = numbering_removed.strip()
9
10 # menghilangkan tanda baca
11 punctuation_removed = whitespace_removed.translate(str.maketrans(
    ("", "", string.punctuation)))
```

Kode 4.6: Potongan kode case folding

Potongan Kode 4.6 digunakan untuk melakukan tahap *case folding*

C Tokenisasi Kata

Tahapan tokenisasi kata ini dilakukan terhadap artikel berita sehingga setiap kata dapat dilakukan pengecekan terhadap kamus yang telah dibuat dari dataset.

```
1 #tokenizing kata
2 word_tokenized = nltk.tokenize.word_tokenize(punctuation_removed
    )
```

Kode 4.7: Potongan kode tokenisasi kata

Tahapan ini dibantu dengan *library* nltk, yang memungkinkan penggunaan fungsi `word_tokenize()`.

4.2.4 Jaccard Similarity

Tahapan ini berguna untuk menghitung kesamaan atau similaritas kata yang telah ditokenisasi terhadap kamus yang dimiliki. Pada tahapan ini, terdapat dua buah fungsi yang dibuat, pertama adalah fungsi yang berguna sebagai formula *jaccard index* dan kedua adalah fungsi yang berguna untuk mengecek apakah token kata yang diterima dari artikel harus dihitung atau tidak. Dari formula *Jaccard Similarity* yang digunakan untuk menghitung similaritas dua buah kata, maka diterjemahkanlah ke dalam bentuk kode sebagai berikut :

```
1 #Formula jaccard yang digunakan untuk menghitung similarity setiap
   kata .
2 def jaccard_similarity(list1 , list2):
3     s1 = set(list1)
4     s2 = set(list2)
5     return float(len(s1.intersection(s2)) / len(s1.union(s2)))
```

Kode 4.8: Potongan kode formula jaccard similarity

Potongan Kode 4.8 menunjukkan bahwa fungsi tersebut membutuhkan dua buah parameter yaitu `list1` dan `list2`, hal ini diperlukan karena formula *jaccard index* membandingkan huruf dalam dua buah kata.

```
1 def hitung_similarity(hasil_preproses , data):
2     y = 0
3     factory = StemmerFactory()
4     stemmer = factory.create_stemmer()
5     for list_words in hasil_preproses:
6         y = y + 1
7         if stemmer.stem(list_words)[0] in CONSTANT:
8             for data_words in data:
9                 if list_words[0] == data_words[0]:
10                     if abs(len(list_words) - len(data_words)) <= 1:
11                         if stemmer.stem(list_words) == stemmer.stem(
12                             data_words):
13                             x = jaccard_similarity([*data_words], [*list_words
14 ])
15                             if x == 1:
16                                 print('test similarity kata', list_words ,': ',
17 x)
```

```

15         print('Kata', list_words , 'sesuai dengan kata',
data_words)
16         tn = tn + 1
17         elif x <= 0.99:
18             print('test similarity kata', list_words ,': ',
x)
19             print('Kata', list_words , 'tidak sesuai dengan
kata', data_words)
20             else :
21                 pass
22             else :
23                 pass
24         else :
25             pass
26     else :
27         pass

```

Kode 4.9: Potongan kode pengecekan dan perhitungan jaccard similarity

Potongan Kode 4.9 menunjukkan bahwa fungsi `hitung_similarity` membutuhkan dua buah parameter yaitu `hasil_preproses` dan `data`, `hasil_preproses` berisikan hasil tokenisasi dari artikel berita dan `data` berupa kamus.

4.3 Uji Coba

Uji coba dilakukan dengan menggunakan *dataset* dari Tribun yang berupa artikel atau berita, pengecekan dilakukan terhadap kamus yang berisikan 252 kata terluluh. Pengujian dilakukan secara bertahap, dari kalimat dengan kata terluluh benar, kata terluluh salah, 2 kalimat, 4 kalimat dan 6 kalimat.

4.3.1 Uji Coba Benar

Pada uji coba pertama, kalimat diambil dari potongan berita yang mengandung kata terluluh. Kalimat terhitung memiliki 30 kata dan 247 karakter. Potongan berita dapat dilihat pada Gambar 4.3.

Selanjutnya, penyidik Polda Metro Jaya akan menyerahkan tersangka dan alat bukti kasus pelanggaran kekarantinaan kesehatan yang melibatkan Rachel Vennya, Salim Nauderer, Maulida Khairunnisa dan seorang petugas bandara Soekarno-Hatta berinisial OP.



Gambar 4.3. Potongan berita benar

Berdasarkan potongan kalimat berita seperti di Gambar 4.3. Didapatkanlah hasil bahwa berita memiliki dua kata terluluh yaitu penyidik dan menyerahkan, serta menyatakan bahwa hasil perhitungan similaritasnya adalah 1.0 yang berarti kata yang dituliskan benar atau sesuai. Terdapat token kata yang tidak memiliki hasil perhitungan, hal ini disebabkan karena token tersebut tidak sesuai dengan kriteria yang ditentukan untuk diperhitungkan. Kriteria-kriteria yang ditentukan adalah sebagai berikut :

1. Huruf pertama pada token setelah kata diubah menjadi kata dasar adalah s, p, t, atau k.
2. Huruf pertama pada token kata dan huruf pertama pada kata pada kamus sama, contohnya adalah dipercayai pada artikel tidak akan dicek dengan kata memercayai dari kamus.
3. Jumlah huruf pada kata yang telah ditoken dan jumlah huruf pada kata dari kamus tidak berjarak lebih dari 1.
4. Kata dasar pada token harus sama dengan kata dasar kata pada kamus.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

```

selanjutnya
penyidik
test similarity kata penyidik : 1.0
Kata penyidik sesuai dengan kata penyidik
polda
metro
jaya
menyerahkan
test similarity kata menyerahkan : 1.0
Kata menyerahkan sesuai dengan kata menyerahkan
tersangka
alat
bukti
kasus
pelanggaran
kekarantinaaan
kesehatan
melibatkan
rachel
vennya
salim
nauderer
maulida
khairunnisa
seorang
petugas
bandara
soekarnohatta
berinisial
op

```

Gambar 4.4. Hasil uji coba potongan berita benar

4.3.2 Uji Coba Kesalahan

Pada uji coba kedua, kalimat diambil dari potongan berita yang mengandung kata terluluh yang benar dan salah. Kalimat terhitung memiliki 2 kalimat, 24 kata dan 158 karakter. Potongan berita dapat dilihat pada Gambar 4.5.

Titi Kamal menyebutkan jika sosok suaminya adalah laki-laki yang setia. Pemain film Ada Apa Dengan Cinta ini mengaku jika dirinya amat mempercayai sang suami.

Gambar 4.5. Potongan berita salah

Berdasarkan potongan kalimat berita di atas. Didapatkanlah hasil bahwa berita memiliki dua kata terluluh yaitu menyebutkan dan mempercayai, serta menyatakan bahwa hasil perhitungan similaritas kata menyebutkan adalah 1.0 yang berarti kata yang dituliskan benar atau sesuai dan hasil perhitungan similaritas kata

mempercayai adalah 0.875, hal ini menyatakan bahwa kata mempercayai adalah kata yang peluluhanannya tidak tepat.

```
titi
kamal
menyebutkan
test similarity kata menyebutkan : 1.0
Kata menyebutkan sesuai dengan kata menyebutkan
sosok
suaminya
lakilaki
setia
pemain
film
ada
apa
dengan
cinta
mengaku
dirinya
mempercayai
test similarity kata mempercayai : 0.875
Kata mempercayai tidak sesuai dengan kata mempercayai
sang
suami
```

Gambar 4.6. Hasil uji coba potongan berita salah

4.3.3 Uji Coba Dengan Parameter

Pada uji coba ketiga, uji coba akan dilakukan dengan parameter sebagai berikut :

Tabel 4.2. Parameter uji coba

Jumlah Kalimat	Jumlah Kata
2	25
2	44
4	55
4	95
6	124
6	148

A Uji Coba Parameter 2 Kalimat

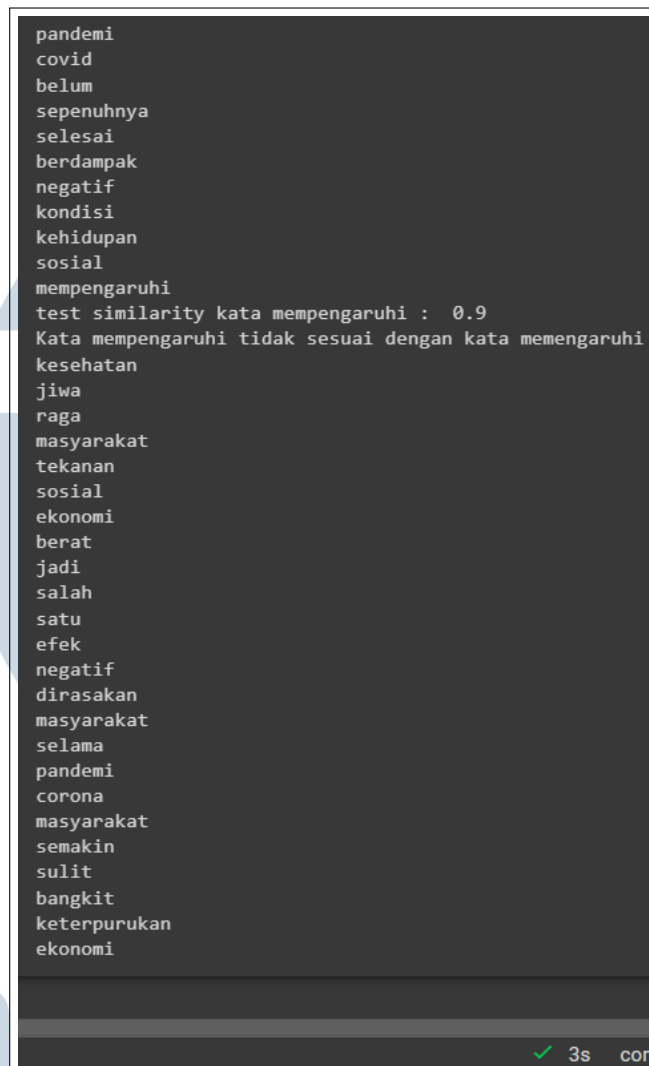
Pada uji coba ini, digunakan 2 kalimat dengan jumlah kata sebanyak 25 kata dan 44 kata, hasil yang didapatkan adalah seperti pada Gambar 4.7 dan Gambar 4.8.

```
belum
keputusan
pbnu
kata
imam
lewat
pesan
singkat
diterima
jumat
lebih
lanjut
imam
mengatakan
test similarity kata mengatakan : 1.0
Kata mengatakan sesuai dengan kata mengatakan
pihaknya
bakal
mengkonfirmasi
test similarity kata mengkonfirmasi : 0.9090909090909091
Kata mengkonfirmasi tidak sesuai dengan kata mengonfirmasi
soal
perintah
tersebut
```

✓ 13s complet

Gambar 4.7. Hasil uji coba parameter 2 kalimat 25 kata

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 4.8. Hasil uji coba parameter 2 kalimat 44 kata

Gambar 4.7 menunjukkan bahwa terdeteksi 2 buah kata terluluh dan deteksi dilakukan dengan benar karena kata mengkonfirmasi adalah kesalahan sehingga perhitungan *jaccard similarity*-nya bernilai kurang dari 1. Berdasarkan data yang diberikan, sistem dapat mendeteksi kesalahan dalam waktu 13 detik. Gambar 4.8 mendeteksi bahwa terdapat 1 buah kesalahan yaitu kata mempengaruhi dengan hasil perhitungan 0.9. Sistem dapat mendeteksi kesalahan dalam waktu 3 detik.

B Uji Coba Parameter 4 Kalimat

Pada uji coba ini, digunakan 4 kalimat dengan jumlah kata sebanyak 55 kata dan 95 kata, hasil yang didapatkan adalah seperti pada Gambar 4.9 dan Gambar

4.10.

```
test similarity kata menanyakan : 1.0
Kata menanyakan sesuai dengan kata menanyakan
test similarity kata menyangka : 1.0
Kata menyangka sesuai dengan kata menyangka
test similarity kata menyangka : 1.0
Kata menyangka sesuai dengan kata menyangka
test similarity kata mengatakan : 1.0
Kata mengatakan sesuai dengan kata mengatakan
test similarity kata menunjukan : 1.0
Kata menunjukan sesuai dengan kata menunjukkan
```

✓ 12s

Gambar 4.9. Hasil uji coba parameter 4 kalimat 55 kata

```
test similarity kata mensosialisasikan : 0.9
Kata mensosialisasikan tidak sesuai dengan kata menyosialisasikan
test similarity kata penerapan : 1.0
Kata penerapan sesuai dengan kata penerapan
test similarity kata penerapan : 1.0
Kata penerapan sesuai dengan kata penerapan
test similarity kata menuju : 0.8333333333333334
Kata menuju tidak sesuai dengan kata menuju
```

✓ 10s completed at 3:3

Gambar 4.10. Hasil uji coba parameter 4 kalimat 95 kata

Gambar 4.9 menunjukkan bahwa terdapat 5 kata terluluh yang mendapatkan nilai 1, hal ini menyatakan bahwa peluluhan telah sesuai, namun terdapat sebuah kesalahan perhitungan pada kata menunjukan, hal ini terjadi karena formula *jaccard similarity* menggunakan *intersection* atau irisan sehingga saat perhitungan walaupun kehilangan 1 huruf k maka akan tetap dihitung *similar* atau sama. Sistem dapat mendeteksi kesalahan dalam waktu 12 detik. Gambar 4.10 menunjukkan bahwa terdapat 4 kata terluluh dengan 2 kata salah dan 2 kata benar dengan waktu deteksi selama 10 detik.

C Uji Coba Parameter 6 Kalimat

Pada uji coba ini, digunakan 6 kalimat dengan jumlah kata sebanyak 124 kata dan 148 kata, hasil yang didapatkan adalah seperti pada Gambar 4.11 dan Gambar 4.12.

```
test similarity kata memastikan : 1.0
Kata memastikan sesuai dengan kata memastikan
test similarity kata mensukseskan : 0.875
Kata mensukseskan tidak sesuai dengan kata menyukseskan
test similarity kata menanggapi : 1.0
Kata menanggapi sesuai dengan kata menanggapi
test similarity kata menanggapi : 1.0
Kata menanggapi sesuai dengan kata menanggapi
test similarity kata mengatakan : 1.0
Kata mengatakan sesuai dengan kata mengatakan
test similarity kata memahami : 1.0
Kata memahami sesuai dengan kata memahami
```

✓ 25s com

Gambar 4.11. Hasil uji coba parameter 6 kalimat 124 kata

```
test similarity kata mengatakan : 1.0
Kata mengatakan sesuai dengan kata mengatakan
test similarity kata mentekankan : 0.8333333333333334
Kata mentekankan tidak sesuai dengan kata menekankan
test similarity kata menyarankan : 1.0
Kata menyarankan sesuai dengan kata menyarankan
test similarity kata menerbitkan : 1.0
Kata menerbitkan sesuai dengan kata menerbitkan
```

✓ 18s co

Gambar 4.12. Hasil uji coba parameter 6 kalimat 148 kata

Gambar 4.11 menunjukkan bahwa terdapat 1 kata terdeteksi salah, 1 kata tidak terdeteksi salah dan 4 kata terdeteksi benar. Sistem dapat mendeteksi 6 kata terluluh dalam waktu 25 detik. Gambar 4.12 menunjukkan 1 kata terdeteksi salah, 1 kata tidak terdeteksi salah dan 2 kata terdeteksi benar. Sistem mendeteksi 4 kata dalam waktu 18 detik.

4.3.4 Perhitungan Confusion Matrix

Perhitungan *confusion matrix* dilakukan dengan data yang didapatkan dari hasil uji coba menggunakan 10 artikel yang terdiri dari 175 kalimat, 3174 kata, dan 23184 karakter. *Confusion matrix* memiliki 4 buah nilai yaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative*. Pada penelitian ini nilai TP (*True Positive*) adalah kata yang salah dan dinilai kurang dari 1 oleh sistem, TN (*True Negative*) adalah kata yang benar dan dinilai 1 oleh sistem, FP (*False Positive*) adalah kata yang salah namun dinilai 1 oleh sistem, dan FN (*False Negative*) adalah kata yang benar namun dinilai kurang dari 1 oleh sistem.

		ACTUAL VALUES	
		POSITIF	NEGATIF
PREDICTED VALUE	POSITIF	6	6
	NEGATIF	0	77

Gambar 4.13. Confusion matrix hasil uji coba

Dari Gambar 4.13 dapat ditentukan bahwa :

- TP = 6
- TN = 77
- FP = 6
- FN = 0

4.4 Evaluasi

Evaluasi pada penelitian ini dibagi menjadi dua, yaitu evaluasi menggunakan *confusion matrix* dan evaluasi terhadap efisiensi sistem.

4.4.1 Evaluasi Confusion Matrix

Hasil *confusion matrix* dari uji coba, hasil tersebut dapat digunakan untuk melakukan perhitungan *accuracy*, *precision*, dan *recall*. Uji coba dilakukan dengan menggunakan 10 artikel secara langsung didapatkan nilai berupa TP sejumlah 6 kata, TN sejumlah 77 kata, FP 6 kata, dan FN 0 kata. Berdasarkan formula yang telah dimiliki, berikut adalah hasil perhitungannya.

1. *Accuracy*

$$accuracy = \frac{6 + 77}{6 + 77 + 6 + 0} \quad (4.1)$$

$$accuracy = \frac{83}{89}$$

$$accuracy = 0.932 * 100\%$$

$$accuracy = 93.2\%$$

2. *Precision*

$$precision = \frac{6}{6 + 6} \quad (4.2)$$

$$precision = \frac{6}{12}$$

$$precision = 0.5 * 100\%$$

$$precision = 50\%$$

3. *Recall*

$$recall = \frac{6}{6 + 0} \quad (4.3)$$

$$recall = \frac{6}{6}$$

$$recall = 1 * 100\%$$

$$recall = 100\%$$

4. *F-1 Score*

$$F-1score = \frac{2 * 1 * 0.5}{0.5 + 1} \quad (4.4)$$

$$F-1score = \frac{1}{1.5}$$

$$F-1score = 0.66 * 100\%$$

$$F-1score = 66\%$$

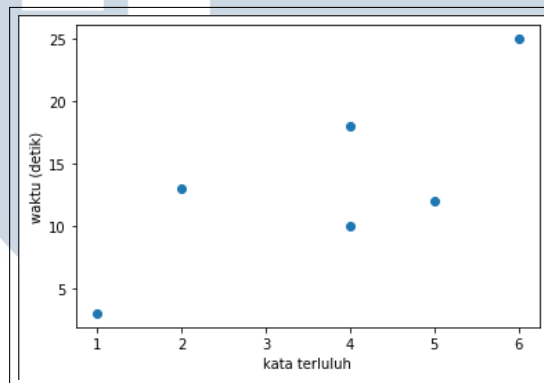
4.4.2 Evaluasi Efisiensi Sistem

Hasil uji coba dengan parameter menunjukkan berapa waktu yang diperlukan oleh sistem untuk menyelesaikan pendeteksian kata terluluh. Hasil uji coba efisiensi sistem dapat dilihat pada Gambar 4.14.

Parameter ke-	Jumlah kata terluluh	Jaccard similarity = 1	Jaccard similarity < 1	Waktu(detik)
1	2	1	1	13
2	1	0	1	3
3	5	5	0	12
4	4	2	2	10
5	6	5	1	25
6	4	3	1	18

Gambar 4.14. Evaluasi uji coba efisiensi

Berdasarkan evaluasi uji coba efisiensi, dibuatlah grafik untuk memvisualisasikan hasil evaluasinya.



Gambar 4.15. Grafik uji coba efisiensi

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

BAB 5

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil uji coba dan evaluasi yang dilakukan, dapat disimpulkan bahwa deteksi kesalahan eja pada kata luluh telah berhasil dibangun dengan menggunakan algoritma *Jaccard Similarity*. Melalui tahap uji coba, didapatkan hasil perhitungan yang menunjukkan bahwa sistem memiliki tingkat *F-1 Score* pada angka 66.6%. Hal ini menunjukkan bahwa sistem mampu mendeteksi namun masih terdapat kesalahan dalam mendeteksi kesalahan eja pada kata terluluh. Selain itu waktu pengerjaan yang dilakukan oleh sistem dipengaruhi oleh jumlah kata terluluh, semakin banyak kata terluluhnya maka waktu yang dibutuhkan oleh sistem dalam mendeteksi akan semakin lama.

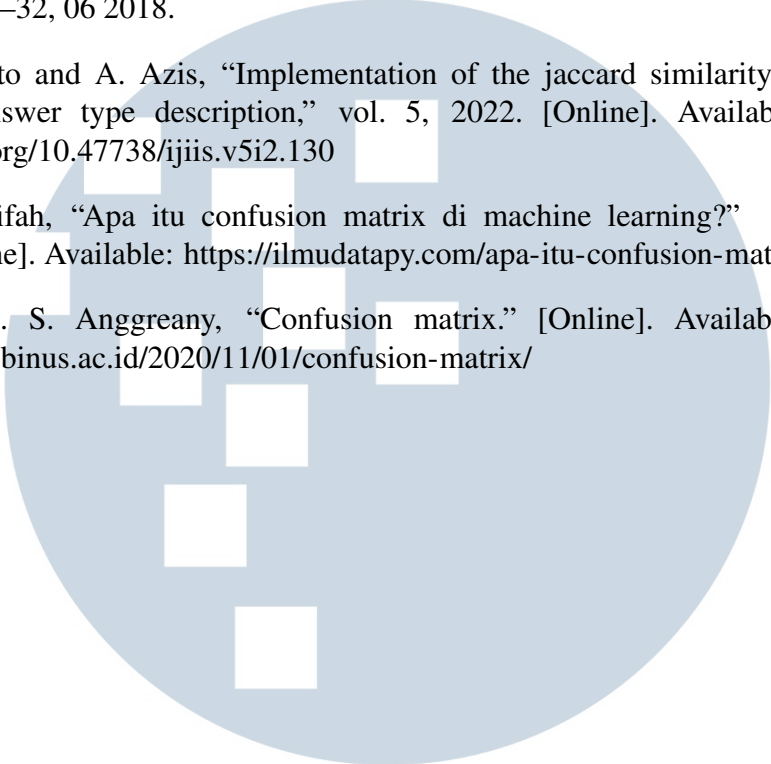
5.2 Saran

Pendeteksian kesalahan eja menggunakan algoritma *Jaccard Similarity* yang dibuat masih terdapat kekurangan dalam proses perhitungannya, terdapat kata-kata salah ketik yang tidak terbaca saat proses *stemming*. Melalui penelitian yang telah dilakukan, proses perhitungan masih dapat dikembangkan seperti menggabungkan dengan *machine learning* untuk mengoreksi kesalahan ketik dan mengubah kata tidak baku menjadi baku terlebih dahulu sebelum diproses sehingga kata yang masuk ke dalam pendeteksian dalam kondisi yang sesuai.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

DAFTAR PUSTAKA

- [1] T. Wiratno and R. Santosa, "Pengantar linguistik umum," *Tangerang Selatan: Universitas Terbuka*, 2011. [Online]. Available: <http://repository.ut.ac.id/4240/1/BING4214-M1.pdf>
- [2] S. Jasmani, "Analisis kesalahan berbahasa indonesia pada berita di portal berita *Online* tribunnews.com," 2021. [Online]. Available: https://digilibadmin.unismuh.ac.id/upload/19199-Full_Text.pdf
- [3] P. Vanya Karunia Mulia, "10 pengertian berita menurut para ahli," 2022. [Online]. Available: <https://www.kompas.com/skola/read/2022/01/06/090000869/10-pengertian-berita-menurut-para-ahli?page=all#:~:text=Djuraid,disampaikan%20oleh%20wartawan%20media%20massa>
- [4] G. Thabroni, "Analisis kesalahan berbahasa : Pengertian, jenis, langkah, dsb," 2022. [Online]. Available: <https://serupa.id/analisis-kesalahan-berbahasa-pengertian-jenis-langkah-dsb/>
- [5] A. Sodikin, "Peluluhan kata dasar berawalan kpst," 2021. [Online]. Available: <https://edukasi.kompas.com/read/2021/01/08/144019571/peluluhan-kata-dasar-berawalan-kpst?page=all>
- [6] S. Niwattatnakul, J. Singthongcai, E. Naenudorn, and S. Wanapu, "Using of jaccard coefficient for keywords similarity." [Online]. Available: https://www.iaeng.org/publication/IMECS2013/IMECS2013_pp380-384.pdf
- [7] [Online]. Available: https://littleflowercollege.edu.in/upload/pdf_upload/f362961fd8c4f41c3defd7ef2ea525aa.pdf
- [8] T. NEWS, "Tribunnews.com - berita terkini indonesia," 2019. [Online]. Available: <https://www.tribunnews.com/>
- [9] I. C. Education, "Natural language processing (nlp)," 2020. [Online]. Available: <https://www.ibm.com/cloud/learn/natural-language-processing>
- [10] K. Poelmans, "What is natural language processing (nlp)?" 2020. [Online]. Available: <https://www.textmetrics.com/what-is-natural-language-processing-nlp>
- [11] R. Hans, "Tahapan text preprocessing dalam teknik pengolahan data," Jun 2021. [Online]. Available: <https://www.dqlab.id/tahapan-text-preprocessing-dalam-teknik-pengolahan-data>
- [12] K. S. Nugroho, "Confusion matrix untuk evaluasi model pada supervised learning," 2020. [Online]. Available: <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine%-learning-bc4b1ae9ae3f>

- 
- [13] A. Kadhim, “An evaluation of preprocessing techniques for text classification,” *International Journal of Computer Science and Information Security*, vol. 16, pp. 22–32, 06 2018.
- [14] Riyanto and A. Azis, “Implementation of the jaccard similarity algorithm on answer type description,” vol. 5, 2022. [Online]. Available: <https://doi.org/10.47738/ijis.v5i2.130>
- [15] L. Afifah, “Apa itu confusion matrix di machine learning?” Sep 2022. [Online]. Available: <https://ilmudatapy.com/apa-itu-confusion-matrix/>
- [16] D. M. S. Anggreany, “Confusion matrix.” [Online]. Available: <https://socs.binus.ac.id/2020/11/01/confusion-matrix/>



Lampiran 1. Form Bimbingan

FORMULIR KONSULTASI SKRIPSI – FAKULTAS TEKNIK & INFORMATIKA


Dosen Pembimbing : Marlinda Vasty Overbeek, S.Kom, M.Kom

Jurusan : Informatika









Semester : 9

Nama : Nicholas Evan

NIM : 00000027900




UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tanggal Konsultasi	Agenda/Pokok Bahasan	Saran Perbaikan	Paraf Dosen Pembimbing
8 Agustus 2022	Pembahasan topik		
26 Agustus 2022	Menentukan metode yang akan digunakan		
9 September 2022	Update progress penelitian terkait crawling data		
28 September 2022	Update progress penelitian terkait preprocessing		
2 Desember 2022	Diskusi tentang metode yang digunakan	Mengubah metode menjadi algoritma jaccard similarity	
13 Desember 2022	Review progress		
20 Desember 2022	Review progress dan laporan		
27 Desember 2022	Review progress dan laporan		

Catatan : Form ini wajib dibawa pada saat konsultasi & dilampirkan didalam skripsi (**Minimal 8 kali Konsultasi**)

Tangerang, 2 Januari 2023



Marlinda Vasty Overbeek, S.Kom, M.Kom

Kampus UMN, Scientia Garden | Jl. Boulevard Gading Serpong – Tangerang | P. +62 21 5422 0808 | F. +62 21 5422 0800 | www.umn.ac.id

Lampiran 2. Uji Turnitin

