PERBANDINGAN METODE *LEMMATIZATION* DAN *STEMMING* TERHADAP PENILAIAN JAWABAN PENDEK OTOMATIS MENGGUNAKAN TF-IDF DAN *COSINE SIMILARITY* TEKS BAHASA INDONESIA

TUGAS AKHIR

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1)
di Program Studi Teknik Informatika, Jurusan Teknologi,
Produksi dan Industri, Institut Teknologi Sumatera

Oleh:

EDINIA ROSA FILIANA

119140018

ITERA

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI, PRODUKSI DAN INDUSTRI

INSTITUT TEKNOLOGI SUMATERA

LAMPUNG SELATAN

2022

LEMBAR PENGESAHAN

Tugas Akhir dengan judul "Perbandingan Metode *Lemmatization* dan *Stemming* Terhadap Penilaian Jawaban Pendek Otomatis Menggunakan TF-IDF dan *Cosine Similarity*" adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya, baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Institut Teknologi Sumatera maupun di institusi pendidikan lainnya.

Lampung Selatan,
Penulis,

PHOTO BERWARNA

Edinia Rosa Filiana
NIM. 119140018

Diperiksa dan disetujui oleh,

Pembimbing	Tanda Tangan
1. Winda Yulita, S.Pd., M.Cs NIP. 19930727 2022 03 2 022	……
2. Mugi Praseptiawan, S.T., M.Kom NIP. 19850921 201903 1 012	……
Penguji	**Tanda Tangan**
1. Ilham Firman Ashari, S.Kom., M.T NIP. 19930314 201903 1 018	……
2. Eko Dwi Nugroho, S.Kom., M.Cs NRK. 1991020 2020 1 279	……

Disahkan oleh,
Koordinator Program Studi Teknik Informatika
Jurusan Teknologi, Produksi dan Industri
Institut Teknologi Sumatera

Andhika Setiawan, S.Kom., M.Cs
NIP. 19911127 2022 03 1 007

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir dengan judul "PERBANDINGAN METODE *LEMMATIZATION* DAN *STEMMING* TERHADAP PENILAIAN JAWABAN PENDEK OTOMATIS MENGGUNAKAN TF-IDF DAN *COSINE SIMILARITY* TEKS BAHASA INDONESIA" adalah karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan benar.

Nama : Edinia Rosa Filiana

NIM : 119140018

Tanda Tangan : …….

Tanggal : …….

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Institut Teknologi Sumatera, saya yang bertanda tangan di bawah ini:

Nama	: Edinia Rosa Filiana
NIM	: 119140018
Program Studi	: Teknik Informatika
Jurusan	: Jurusan Teknologi, Produksi dan Industri
Jenis Karya	: Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Sumatera **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah saya yang berjudul:

PERBANDINGAN METODE *LEMMATIZATION* DAN *STEMMING* TERHADAP PENILAIAN JAWABAN PENDEK OTOMATIS MENGGUNAKAN TF-IDF DAN *COSINE SIMILARITY* TEKS BAHASA INDONESIA beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini, Institut Teknologi Sumatera berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Lampung Selatan
Pada tanggal 17 September 2022

Yang menyatakan,

Edinia Rosa Filiana

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga penyusunan tugas akhir ini telah terselesaikan dengan baik. Dalam penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapan terima kasih kepada:

1. <isi dengan nama Rektor ITERA>
2. <isi dengan nama Kajur JTPI>
3. <isi dengan nama Kaprodi IF>
4. <isi dengan nama Sesprodi IF>
5. <isi dengan nama Koordinator TA>
6. <isi dengan nama Dosen Pembimbing>
7. Kedua Orang Tua, kakak dan adik yang selalu memberikan arahan selama belajar dan menyelesaikan tugas akhir ini.
8. <isi dengan nama orang lainnya>

Akhir kata penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi kita semua, amin. [Contoh]

RINGKASAN

PERBANDINGAN METODE *LEMMATIZATION* DAN *STEMMING* TERHADAP PENILAIAN JAWABAN PENDEK OTOMATIS MENGGUNAKAN TF-IDF DAN *COSINE SIMILARITY* TEKS BAHASA INDONESIA

Edinia Rosa Filiana

Halaman Ringkasan berisi uraian singkat tentang latar belakang masalah, rumusan masalah, tujuan, metodologi penelitian, hasil dan analisis data, serta kesimpulan dan saran. Isi ringkasan tidak lebih dari 1500 kata (sekitar 3 halaman).

ABSTRAK

PERBANDINGAN METODE *LEMMATIZATION* DAN *STEMMING* TERHADAP PENILAIAN JAWABAN PENDEK OTOMATIS MENGGUNAKAN TF-IDF DAN *COSINE SIMILARITY* TEKS BAHASA INDONESIA

Edinia Rosa Filiana

Halaman ABSTRAK berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INDONESIA tidak lebih dari 250 kata, dengan jarak antar baris satu spasi.

Pada akhir abstrak ditulis kata “Kata Kunci” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Kata kunci terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Kata kunci diketik pada jarak dua spasi dari baris akhir isi abstrak.

Kata Kunci : Penambangan Data, Kecerdasan Buatan, Lampung Selatan

ABSTRACT

COMPARISON OF LEMMATIZATION AND STEMMING ON INDONESIAN AUTOMATIC SHORT ANSWER GRADING USING TF-IDF AND COSINE SIMILARITY

Edinia Rosa Filiana

Halaman ABSTRACT berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INGGRIS tidak lebih dari 250 kata, dengan jarak antar baris satu spasi. Secara khusus, kata dan kalimat pada halaman ini tidak perlu ditulis dengan huruf miring meskipun menggunakan Bahasa Inggris, kecuali terdapat huruf asing lain yang ditulis dengan huruf miring (misalnya huruf Latin atau Greek, dll).

Pada akhir abstract ditulis kata “Keywords” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Keywords terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Keywords diketik pada jarak dua spasi dari baris akhir isi abstrak.

Keywords : Data Mining, Artificial Intelligence, Lampung Selatan

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR RUMUS

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Berkembangnya teknologi seiring jaman menuntut manusia untuk dapat mengikuti perkembangannya. Penerapan teknologi tersebut tersebar dari berbagai segi mulai dari segi ekonomi, industri, pendidikan, dan masih banyak lagi. Perkembangan tersebut diiringi dengan kebutuhan dari masyarakat dan menuntut para peneliti untuk menemukan inovasi dari masalah dan kebutuhan baru dalam masyarakat. Salah satu kebutuhan baru juga muncul berasal dari sektor pendidikan. Dilansir dari data Badan Pusat Statistik, jumlah guru dan murid pada jenjang Sekolah Dasar (SD), Sekolah Menengah Pertama (SMP), Sekolah Menengah Atas (SMA), dan Perguruan Tinggi memiliki perbandingan yang cukup besar. Total keseluruhan murid di Indonesia mencapai angka 47.598.980. Sedangkan total jumlah guru yang ada terhitung hanya mencapai 2.953.325 [1], [2], [3], [4]. Grafik perbandingan jumlah guru dan murid Indonesia pada Tahun 2021 dapat dilihat pada Gambar 1.1.

Gambar 1.1. Perbandingan Jumlah Guru dan Murid Indonesia Tahun 2021

Berdasarkan Gambar 1.1, dapat dilihat bahwa persentase guru hanya 5,8% berbanding jauh dengan murid yang mencapai 94,2%. Hal tersebut menunjukkan kesenjangan yang besar antara jumlah pendidik dan siswa Indonesia.

Guru atau pengajar memiliki banyak tugas lain disamping mengajar dalam kelas. Salah satunya adalah melakukan penilaian atau evaluasi kemampuan siswa hasil dari

proses belajar mengajar yang telah di lakukan dengan metode ujian. Ujian pada umumnya memiliki tipe soal objektif seperti pilihan ganda dan/atau subjektif seperti esai [5]. Soal esai memiliki kelebihan dalam hal pengukuran kemampuan siswa dalam mendalami kompetensi pembelajaran. Hal tersebut dikarenakan jawaban dari soal esai berbentuk penjelasan dalam beberapa kata maupun kalimat yang disusun sendiri berdasarkan daya ingat dari siswa [6].

Namun, jenis soal esai juga memiliki permasalahan dalam proses pengoreksiannya. Waktu yang dibutuhkan dalam pengoreksian soal uraian seperti esai lebih besar dibandingkan dengan soal pilihan ganda yang sudah jelas benar salahnya. Manusia mudah dipengaruhi oleh subjektifitas, sehingga jika pengoreksi terdiri dari beberapa orang yang berbeda maka nilai yang diberikan juga akan berbeda. Selain itu, kemampuan manusia juga acap kali dipengaruhi oleh kondisi fisik dan mental sehingga pemberian nilai juga menjadi inkonsisten bergantung pada kondisi pengoreksi saat ini [6]. Inkonsisten dan subjektifitas tersebut sangat rawan terjadi jika pengoreksi dituntut untuk mengoreksi dalam waktu singkat namun jumlah ujian yang harus dikoreksi banyak.

Terdapat beberapa aplikasi yang telah diterapkan untuk memecahkan permasalahan tersebut seperti *Project Essay Grader* (PEG), e-Rater, *Bayesian Essay Test Scoring System*, lalu SIMPLE atau Sistem Penilaian Esai Otomatis yang dikembangkan untuk menilai ujian dalam bahasa Indonesia [7]. Selain metode-metode di tersebut, terdapat metode lain seperti *Cosine Similarity*. Metode tersebut dapat dipakai untuk menghitung tingkat kemiripan suatu teks atau dokumen dengan teks atau dokumen lain dengan membobotkan setiap kata pada dokumen teks tersebut.

Pembobotan kata dengan menyamakan frekuensi biasa disebut dengan *Term Frequency*. Penentuan bobot kata dapat dihitung dengan menghitung banyaknya kemunculan kata yang dimaksud pada dokumen. Salah satu metode Term Frequency adalah metode *Term Frequency Inverse-Document Frequency* (TF-IDF). Menurut penelitian yang dilakukan oleh Musfiroh Nurjannah, penggunaan TF-IDF sangat membantu dalam proses pencarian informasi pada kumpulan dokumen [8]. Metode ini merupakan metode untuk membobotkan kata berdasarkan hubungan kata tersebut dalam suatu dokumen dengan menggabungkan konsep perhitungan frekuensi

kemunculan suatu kata dengan *inverse* frekuensi dokumen yang memiliki kata yang dimaksud [9].

Sebelum teks dapat dilakukan pembobotan, perlu adanya tahap *pre-processing*. Teks *pre-processing* adalah salah satu tahap untuk membersihkan data yang berbentuk teks dalam *text mining* untuk mengurangi inkonsistensi data [10]. Oleh karena itu, tahap ini akan sangat mempengaruhi hasil akurasi dari pembobotan. Umumnya tahap pada teks *pre-processing* ini terdiri dari proses case *folding*, *tokenization*, *filtering*, dan *stemming*, *lemmatization*. *Stemming* dan *lemmatization* adalah dua prosedur yang biasanya digunakan untuk membantu menyempurnakan model dalam pengolahan bahasa dengan mempercepat proses pencarian informasi [11]. Kedua prosedur ini juga memiliki tujuan yang serupa yaitu menormalkan dan mengembalikan kata menjadi kata dasarnya [12]. Namun, terdapat perbedaan yang cukup mencolok antara kedua metode ini. *Lemmatization* sendiri adalah salah satu proses dalam tahap *pre-processing* untuk menormalisasi varian morfologi yang berbeda dengan menghilangkan prefiks dan sufiks pada kata, lalu mengembalikan kata tersebut ke bentuk dasarnya yang memiliki makna [13]. Sedangkan, *stemming* adalah proses untuk menemukan kata dasar kata dengan memetakan bentuk varian kata dengan menghilangkan imbuhan pada kata sehingga memungkinkan kata tersebut menjadi tidak bermakna [14].

Terdapat beberapa penelitian yang telah membandingkan performa dari kedua metode *lemmatization* dan *stemming* dalam penerapan *information retrieval*. Hasil dari penelitian yang dilakukan oleh Balakhrisnan menunjukkan bahwa metode *lemmatization* memiliki tingkat presisi yang lebih baik dari *stemming* terhadap *information retrieval* walau tidak signifikan [11]. Selain itu, penelitian yang dilakukan oleh Korenius menunjukkan bahwa metode *stemming* mendapatkan nilai *recall* yang lebih tinggi namun presisinya lebih rendah. Namun secara umum, metode *lemmatization* lebih efektif dibandingkan dengan *stemming* karena rata-rata dari nilai *recall* dan presisi yang dihasilkan metode *lemmatization* lebih baik [15]. Selanjutnya ada pula penelitian yang dilakukan oleh Boban terkait *information retrieval* yang didasarkan pada perbedaan panjang *query*. Hasil penelitian menunjukkan bahwa kedua metode sama-sama baik untuk *information retrieval* dengan perbedaan *stemming* lebih baik dari *lemmatization* saat *query* yang digunakan pendek (50 kata), sedangkan *lemmatization* memiliki performa yang lebih baik jika *query* yang digunakan panjang

(lebih dari 39,000 kata) [16]. Untuk penelitian yang membandingkan *lemmatization* dan *stemming* pada Bahasa Indonesia, kedua metode tersebut dibandingkan pada kasus Analisis Sentimen. Penelitian tersebut dilakukan oleh Fadel dan mendapatkan hasil akurasi *lemmatization* mencapai 84% sedangkan *stemming* mendapatkan hasil 85% [17].

Namun, seluruh penelitian terkait pemodelan penilaian esai otomatis baik esai panjang maupun pendek untuk Bahasa Indonesia sejauh ini hanya menggunakan metode *stemming* dalam tahap *pre-processing*nya. Disamping itu, sampai saat ini juga belum ada yang melakukan penelitian untuk membandingkan akurasi atau performa dari metode *lemmatization* dan *stemming* pada pemodelan Penilaian Esai Pendek Otomatis terutama dalam Bahasa Indonesia. Berdasarkan permasalahan tersebut, penulis tertarik untuk melakukan penelitian untuk membandingkan akurasi dari metode *lemmatization* dan *stemming* terhadap pemodelan penilaian esai pendek otomatis teks Bahasa Indonesia menggunakan pembobotan TF-IDF dan *Cosine Similarity*.

1.2 Rumusan Masalah

Berdasarkan penjabaran masalah yang telah diungkapkan pada latar belakang, maka rumusan masalah pada penelitian ini adalah bagaimana tingkat akurasi dari penilaian jawaban pendek otomatis menggunakan metode *Cosine Similarity* dan TF-IDF dengan penerapan *lemmatization* dan *stemming* dalam teks berbahasa Indonesia?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah dan penjabaran latar belakang yang telah diungkap peneliti maka tujuan penelitian ini adalah mendapatkan tingkat akurasi dari penilaian jawaban pendek otomatis menggunakan metode *Cosine Similarity* dan TF-IDF dengan penerapan *lemmatization* dan *stemming* dalam teks berbahasa Indonesia.

1.4 Batasan Masalah

Berdasarkan rumusan masalah dan penjabaran latar belakang sebelumnya, maka batasan masalah yang ada pada penelitian ini adalah sebagai berikut:

1. Dataset yang digunakan adalah *Indonesian Answering Dataset for Online Essay Test System* dengan kategori Teknologi yang disusun oleh Rahutomo [18].
2. Tidak dapat mengolah data yang berada dalam bentuk tabel.
3. Model hanya akan mengolah data berbentuk teks.
4. Model hanya akan mengolah teks berbahasa Indonesia.
5. Menekankan pada penerapan *lemmatization* dan *stemming* pada tahap *pre-processing*.
6. Teks memiliki panjang dengan rentang 3 - 100 kata.

1.5 Manfaat Penelitian

Diharapkan hasil penelitian ini dapat digunakan untuk acuan pemilihan metode dengan akurasi terbaik antara *stemming* dan *lemmatization* untuk dapat diterapkan pada tahap teks *pre*-processing dalam pemodelan penilaian jawaban pendek otomatis.

1.6 Sistematika Penulisan

1.6.1 Bab I Pendahuluan

Pendahuluan berisikan latar belakang masalah, rumusan masalah, batasan penelitian, pemanfaatan penelitian, dan sistematika penulisan.

1.6.2 Bab II Tinjauan Pustaka

Tinjauan pustaka berisikan tinjauan pustaka, serta dasar teori berkaitan dengan materi penunjang penelitian.

1.6.3 Bab III Metode Penelitian

Metode penelitian berisikan metodologi dan alur terkait penelitian yang dilakukan peneliti.

1.6.4 Bab IV Hasil Penelitian dan Pembahasan

Hasil penelitian dan pembahasan berisikan penjelasan dari hasil penelitian mulai dari tahap pengumpulan data, teks *pre-processing*, pembobotan kata, uji similaritas kata, pengujian, perbandingan akurasi dan analisis terhadap hasil pemodelan penilaian

esai otomatis menggunakan metode *Cosine Similarity* dan pembobotan TF-IDF menggunakan penerapan *lemmatization* dan *stemming* pada teks berbahasa Indonesia.

1.6.5 Bab V Kesimpulan dan Saran

Kesimpulan dan saran berisikan kesimpulan dan saran yang didapatkan dari tahap implementasi dan analisis yang telah dilakukan.

BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Terdapat beberapa penelitian berkaitan dengan penelitian ini dan menjadi acuan pendukung penelitian baik dari segi topik maupun metodenya. Daftar penelitian yang digunakan dapat dilihat pada Tabel 2.1.

Tabel 2.1. Tinjauan Pustaka

No	Penulis (Tahun publish)	Judul	Metode	Hasil	Persamaan	Perbedaan
1	Uswatun Hasanah (2018)	*An Experimental Study of Text Preprocessing Techniques for Automatic Short Answer Grading in Indonesian*	1. Cosine Similarity 2. Stemming (Sastrawi) 3. Mean Absolute Error 4. Pearson Correlation	Tahap pre-processing yang digunakan tidak begitu berdampak pada penilaian esai pendek otomatis karena belum bisa menangani kata semantik.	Penelitian ini memiliki persamaan metode yaitu penggunaan metode Stemming dan Cosine Similarity untuk pengembangan model Penilaian Esai Pendek Otomatis	Perbedaan dari penelitian yang dilakukan adalah penelitian ini menambahkan metode Lemmatization pada tahap pre-processing untuk dibandingkan
2	Ivan Boban (2020)	*Sentence retrieval using Stemming and*	1. TF-ISF 2. Stemming	Hasil penelitian menunjukkan bahwa kedua	Penelitian ini memiliki persamaan	Perbedaan dari penelitian yang

No	Penulis (Tahun publish)	Judul	Metode	Hasil	Persamaan	Perbedaan
		Lemmatization with Different Length of the Queries	(Porter stemmer) 3. Lemmatization 4. Mean Average Precision	metode sama-sama baik untuk information retrieval dengan perbedaan stemming lebih baik dari lemmatization saat query yang digunakan pendek (50 kata), sedangkan lemmatization memiliki performa yang lebih baik jika query yang digunakan panjang (lebih dari 39,000 kata)	pada penggunaan metode lemmatization dan stemming.	dilakukan adalah penelitian ini menggunakan metode TF-IDF dan algoritma stemming yang digunakan adalah Nazief dan Adriani
3	Alfirna Rizqi Lahitani (2022)	*Automated Essay Scoring Menggunakan Cosine Similarity pada Penilaian Esai Multi Soal*	1. Cosine Similarity 2. TF-IDF 3. Case folding 4. Stemming 5. Stopwords	Cosine Similarity mewakili 52% poin yang merupakan nilai asli atas jawaban siswa dan tidak memiliki unsur subjektif. Poin dapat ditambahkan oleh pengajar dari unsur penilaian lain untuk melengkapi skor agar nilai menjadi lebih proporsional.	Penelitian ini memiliki persamaan dalam penggunaan metode Cosine Similarity dan TF-IDF untuk penilaian esai otomatis	Perbedaan dari penelitian yang dilakukan adalah penelitian ini menambahkan metode Lemmatization pada tahap pre-processing untuk dibandingkan
4	Mohammad Alobed (2021)	*A Comparative Analysis of Euclidean, Jaccard and Cosine*	Komparasi antara metode Euclidean, Jaccard, Cosine	Didapat bahwa metode Cosine Similarity memiliki nilai error terendah	Penelitian ini memiliki penjelasan terkait salah satu	Perbedaan dari penelitian yang dilakukan adalah

No	Penulis (Tahun publish)	Judul	Metode	Hasil	Persamaan	Perbedaan
		Similarity Measure and Arabic Wordnet for Automated Arabic Essay Scoring	Similarity, metode uji MAE dan Pearson Coefficient		metode yaitu Cosine Similarity	penelitian ini membandingkan metode Lemmatization dan Stemming
5	Nurul Chamidah, Mayanda Mega Santoni (2021)	Pencocokan Berbasis Kata Kunci pada Penilaian Esai Pendek Otomatis Berbahasa Indonesia	1. Case folding 2. Filtering 3. Stemming 4. Tokenisasi 5. Mean Absolute Error (MAE)	Error yang didapatkan masih cukup besar yaitu 25.47%. Hal tersebut dikarenakan masih banyak perbedaan penilaian karena adanya penggunaan bahasa asing seperti user	Penelitian ini memiliki persamaan metode yaitu metode case folding, filtering, dan tokenisasi	Perbedaan dari penelitian yang dilakukan adalah penelitian ini menambahkan metode Lemmatization pada tahap pre-processing untuk dibandingkan
6	Muhammad Walid Arbiantono (2021)	Pengembangan Aplikasi Assesment Tool Menggunakan Metode Cosine Semantic Similarity Untuk Automatic Scoring Jawaban Tes Uraian Pada Mata Pelajaran Basis Data Di SMKN 1 Surabaya	1. Cosine Similarity 2. TF-IDF 3. Stemming (Nazief)	Tidak ada perbedaan yang signifikan dibandingkan dengan penilaian secara manual oleh evaluator	Penelitian ini memiliki persamaan metode Cosine Simlarity, TF-IDF, dan Stemming untuk membangun sistem Penilaian Otomatis	Perbedaan dari penelitian yang dilakukan adalah penelitian ini menambahkan metode Lemmatization pada tahap pre-processing untuk dibandingkan

No	Penulis (Tahun publish)	Judul	Metode	Hasil	Persamaan	Perbedaan
7	Harry Pribadi (2018)	Implementasi Metode Naïve Bayes Classifier Untuk Aplikasi Filtering Email Spam Dengan Lemmatization Berbasis Web	*Naive Bayes Classifier, Lemmatization*	Kecepatan rata-rata Lemmatization saat memisahkan kata dasar adalah 0.86 detik dan kecepatan Lemmatization berbanding lurus dengan ukuran berkas.	Penelitian ini memiliki persamaan di metode text mining yang menerapkan metode Lemmatization	Perbedaan dari penelitian yang dilakukan adalah penelitian ini diterapkan untuk penilaian jawaban pendek otomatis
8	Fadel Maulana Ichsan (2020)	Pengaruh Penggunaan Stemming dan Lemmatization Terhadap Akurasi Analisis Sentimen	1. Stemming 2. Lemmatization 3. Confusion Matrix 4. K-fold cross validation	Hasil akurasi Stemming mencapai 85% sedangkan Lemmatization menghasilkan akurasi 84%.	Penelitian ini memiliki persamaan pada komparasi metode Stemming dan Lemmatization.	Perbedaan dari penelitian yang dilakukan adalah penelitian ini diterapkan untuk penilaian jawaban pendek otomatis

Penelitian pertama berjudul "*An Experimental Study of Text Preprocessing Techniques for Automatic Short Answer Grading in Indonesian*" yang ditulis oleh Uswatun Hasanah pada tahun 2018 [19]. Permasalahan yang diangkat berupa belum adanya teknik yang tepat pada pengolahan bahasa alami agar dapat meningkatkan performa terutama dalam implementasi Penilaian Esai Pendek Otomatis. Adapun metode yang digunakan untuk memecahkan masalah tersebut adalah dengan menggunakan metode *Cosine Similarity* dan *pre-processing* berupa *Case Folding, Tokenization, Punctuation Removal, Stopword Removal, dan Stemming* menggunakan algoritma Sastrawi. Hasil penelitian dievaluasi menggunakan *Mean Absolute Error* dan *Pearson Correlation* untuk mendapatkan akurasi dari sistem Penilaian Esai Pendek Otomatis menggunakan metode-metode yang ada.

Penelitian kedua berjudul "*Sentence retrieval using Stemming and Lemmatization with Different Length of the Queries*" yang ditulis oleh Ivan Boban pada tahun 2020 [16]. Masalah yang diangkat pada penelitian tersebut adalah pada penelitian terdahulu, dikatakan tahap *pre-proxessing* sangat berguna untuk *document retrieval*. Namun, untuk *sentence retrieval* masih belum terdapat kejelasan apakah tahap *pre-processing* tersebut berguna atau tidak untuk diterapkan, Oleh karena itu, penulis melakukan penelitian untuk mengetahui pengaruh pengimplementasian metode *stemming* dan *lemmatization* terhadap *sentence retrieval*. Hasil penelitian menunjukkann bahwa kedua metode berpengaruh baik ketika diimplementasikan pada *sentence retrieval* dengan *stemming* memiliki hasil yang lebih untuk *query* berukuran pendek dibandingkan dengan *lemmatization.* Sedangkan *lemmatization* memiliki hasil yang lebih baik untuk *query* yang berukuran panjang jika dibandingkan dengan *stemming*.

Penelitian terkait ketiga berjudul "*Automated Essay Scoring* menggunakan *Cosine Similarity* pada Penilaian Esai Multi Soal" yang ditulis oleh Alfirna Rizqi Lahitani pada tahun 2022. Masalah yang diangkat pada penelitian tersebut adalah waktu yang di butuhkan untuk menilai jawaban esai secara manual cukup lama. Oleh karena itu, peneliti menggunakan implementasi dari *Automated Essay Scoring* untuk membantupengoreksian dan pemberian skor menggunakan metode pembobotan TF-IDF dan metode pengukuran menggunakan *Cosine Similarity* pada dokumen esai.

Hasil dari penelitian menunjukkan bahwa 52% poin mewakili dari nilai akhir dibandingkan dengan nilai aslinya berdasarkan hasil perhitungan *Cosine Similarity* [20].

Selanjutnya penelitian keempat memiliki judul "*A Comparative Analysis of Euclidean, Jaccard and Cosine Similarity Measure and Arabic Wordnet for Automated Arabic Essay Scoring*" yang ditulis oleh Mohammad Alobed pada tahun 2021 [21]. Topik tersebut diangkat karena peneliti melihat bahwa penelitian terkait pengukuran semantik kata teks berbahasa Arab masih terbatas. Peneliti melakukan komparasi beberapa metode seperti *Euclidean, Jaccard, dan Cosine Similarity* untuk mencari metode terbaik dalam menghitung similaritas dokumen berbahasa Arab. Hasil penelitian menunjukkan bahwa metode *Cosine Similarity* memiliki nilai error terkecil.

Penelitian kelima berjudul "Pencocokan Berbasis Kata Kunci pada Penilaian Esai Pendek Otomatis Berbahasa Indonesia" yang ditulis oleh Nurul Chamidah pada tahun 2021 [14]. Permasalahan yang diangkat berupa proses evaluasi soal esai yang dilakukan oleh satu orang memiliki kemungkinan untuk terjadinya inkonsistesi jika dalam pengevaluasian dilaksanakan pada waktu yang terpisah. Adapun metode yang digunakan untuk memecahkan masalah tersebut adalah dengan mencocokkan kata kunci pada esai lalu dinilai berdasarkan kata kunci yang sesuai. Metode *pre-processing* yang dipakai adalah *case folding, filtering, stemming,* dan *tokenisasi.* Hasil penelitian dievaluasi menggunakan *Mean Absolute Error* untuk membandingkan selisih nilai sistem dengan nilai uji evaluator dan menunjukkan bahwa masih banyak perbedaan atau selisih nilai antara penilaian evaluasi dan penilaian sistem karena adanya terdapat beberapa penggunaan bahasa asing pada esai seperti kata "*user*".

Penelitian keenam berjudul "Menggunakan Metode *Cosine Semantic Similarity* Untuk *Automatic Scoring* Jawaban Tes Uraian Pada Mata Pelajaran Basis Data Di SMKN 1 Surabaya" yang masih ditulis oleh Muhammad Walid Arbiantono pada tahun 2021 [7]. Masalah yang diangkat pada penelitian tersebut adalah tingginya tuntuan untuk mengikuti perkembangan teknologi untuk dapat mendukung proses pembelajaran. Oleh karena itu, peneliti membangun sebuah aplikasi dengan menggunakan penerapan Penilaian Esai Otomatis menggunakan metode *Cosine Similarity* dan TF-IDF. Hasil evaluasi menggunakan koefisien Kappa memperoleh

hasil 0,41 yang menunjukkan bahwa aplikasi tersebut mendapat predikat *good* berdasarkan tabel kategori nilai Kappa.

Penelitian terkait ketujuh berjudul "Implementasi Metode *Naïve Bayes Classifier* Untuk Aplikasi *Filtering Email Spam* Dengan *Lemmatization* Berbasis Web" yang ditulis oleh Harry Pribadi pada tahun 2018 [13]. Masalah yang diangkat pada penelitian tersebut adalah banyaknya pihak yang menyalahgunakan fitur pengiriman pesan ke banyak penerima dalam waktu singkat melalui email dengan mengirimkan peasn berupa promosi, pornografi, virus, dan hal-hal tidak penting lainnya. Oleh karena itu peneliti membangun aplikasi untuk filtering email spam menggunakan metode *Naïve Bayes* dan *Lemmatization* untuk mengklasifikasikan jenis email dan mengelola teks pada email menjadi kata dasar. Hasil dari penelitian tersebut menunjukkan bahwa akurasi algoritma *Naïve Bayes* memiliki angka yang besar yaitu 90,83%.

Penelitian kesembilan berjudul "Pengaruh Penggunaan *Stemming* dan *Lemmatization* Terhadap Akurasi Analisis Sentimen" yang ditulis oleh Fadel Maulana Ichsan pada tahun 2020 [17]. Permasalahan yang diangkat berupa membandingkan teknik pada pre-processing yaitu *Stemming* dan *Lemmatization* untuk mengetahui pengaruhnya jika diterapkan pada analisis sentimen. Adapun metode yang digunakan adalah dengan mengklasifikasi hasil dari pre-processing kedua metode *Lemmatization* dan *Stemming* dengan menggunakan *Support Vector Machine*. Hasil klasifikasi keduanya akan di evaluasi menggunakan *Confusion Matrix*. Hasil evaluasi kedua metode lalu divalidasi menggunakan *K-fold cross validation* untuk mendapatkan nilai paling optimal yang dihasilkan dari kedua metode tersebut.

Berdasarkan hasil dari tinjauan pustaka dari beberapa peneliti, pembeda penelitian ini dengan penelitian sebelumnya adalah penelitian terdahulu telah membandingkan performa dari penerapan metode *Lemmatization* dan *Stemming* pada kasus Analisis Sentimen, namun belum ada penelitian yang membandingkan penerapan metode *Lemmatization* dan *Stemming* terhadap akurasi Penilaian Jawaban Pendek Otomatis. Hasil dari penelitian ini akan memberikan informasi terkait akurasi dari kedua metode yaitu *Stemming* dan *Lemmatization* yang diterapkan pada model Penilaian Esai Pendek Otomatis teks Bahasa Indonesia.

2.2 Dasar Teori

Penelitian ini menggunakan beberapa istilah dan teori sebagai acuan dalam melakukan penelitian. Adapun teori dan istilah yang digunakan pada penelitian dijabarkan pada subbab di bawah ini.

2.2.1 Penilaian Jawaban Pendek Otomatis

Penilaian Esai Otomatis atau *Automated Essay Grading* (AEG) adalah sebuah aplikasi yang dibangun berdasarkan masalah pada sulitnya mengevaluasi nilai dari jawaban siswa yang memperbolehkan siswa untuk memberikan jawaban dengan rangkaian kata yang sesuai dengan kemampuannya dan pemahamannya sendiri [22]. Penilaian Esai Otomatis dapat dikelompokkan berdasarkan panjang dari esai yaitu Penilaian Esai Otomatis (*Automatic Essay Scoring*) untuk jawaban esai yang terdiri dari 2 paragraf hingga beberapa halaman (150-550 kata) dan Penilaian Jawaban Pendek Otomatis (*Automated Short Answer Grading*) untuk jawaban yang memiliki rentang satu frasa hingga satu paragraf (3-100 kata) [23], [24]. Pada penelitian ini, peneliti berfokus pada penilaian jawaban esai pendek atau Penilaian Jawaban Pendek Otomatis.

Penilaian Esai Pendek Otomatis atau lebih dikenal sebagai *Automatic Short Answer Grading* (ASAG) adalah sebuah model untuk memberikan skor atau nilai pada jawaban untuk soal objektif seperti layaknya seorang evaluator menilai esai [25]. Pada penelitian yang dilakukan oleh Burrows, rata-rata jumlah kata per pertanyaan untuk jawaban pendek adalah 92 kata [24]. Konsep penilaian esai otomatis adalah dengan membandingkan teks jawaban siswa dengan kunci jawaban. Semakin dekat nilai kesamaan antara jawaban dengan kunci jawaban menunjukkan semakin besar skor jawaban tersebut. Pengevaluasian akurasi dilakukan dengan membandingkan hasil penilaian sistem dengan hasil penilaian evaluator [26]. Untuk dapat menganalisa hasil jawaban siswa, diperlukan adanya teknik Pengolahan Bahasa Alami. Terdapat dua jenis teknik yang digunakan dalam Penilaian Esai Pendek Otomatis yaitu teknik pemrosesan linguistik dan teknik statistikal [24].

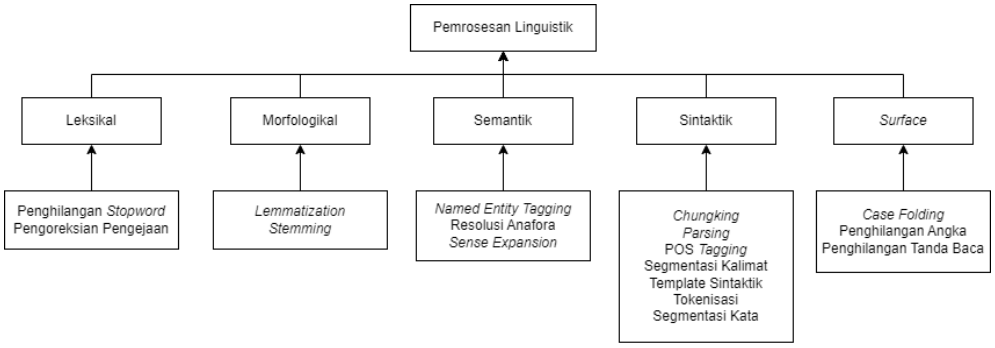
2.2.2 Pengolahan Bahasa Alami

Pengolahan bahasa alami adalah cabang dari Injelensi Buatan (*Artificial Intelligence*) yang dapat membuat komputer dapat membaca, mengerti, dan menginterpretasi bahasa manusia/alami [27]. Sebuah sistem pengolahan bahasa alami harus memperhatikan pengetahuan bahasa yang digunakan, struktur kata pembentuk kalimat, arti kata dan fungsinya dalam sebuah kalimat.

Terdapat beberapa bidang pengetahuan yang ada dalam pengolahan bahasa alami yaitu sebagai berikut [28]:

1. Fonetik dan fonologi, merupakan bidang yang berfokus pada pengenalan suara menjadi kata. Bidang tersebut berperan penting alam proses implementasi aplikasi yang menggunakan metode *speech-based system*.
2. Morfologi, bidang merupakan bidang yang berfokus pada pengetahuan terkait kata dan bentuknya untuk dapat membedakan antara satu kata dengan kata lainnya.
3. Sintaksis, merupakan suatu pemahaman dalam membentuk kalimat dan korelasi antar kata dalam proses prubahan bentuk dari kalimat kedalam bentuk sistematis.
4. Semantik, merupakan pemahaman terkait bentuk struktur sintaksis kat ke dalam bentuk yang lebih mendasar dan tidak bergantung pada struktur kalimat. Mempelajari arti suatu kata dan bagaimana arti kata tersebut membentuk suatu arti dalam kalimat utuh tanpa mencakup konteks dari kalimat.
5. Pragmatik, merupakan pengetahuan berkaitan dengan masing-masing konteks, situasi, dan tujuan pembangunan suatu sistem.
6. *Discourse Knowledge*, melakukan pengenalan terhadap suatu kalimat yang telah dibaca sebelumnya akan mempengaruhi arti dari kalimat selanjutnya. Hal tersebut penting untuk proses pengolahan arti terhadap kata ganti orang serta untuk mengartikan aspek sementara informasi.
7. *World Knowledge*, mencakup arti suatu kata secara umum dan mencari arti khusus bagi suatu kata dalam suatu percakapan dengan konteks tertentu.

Pada penelitian yang dilakukan oleh Burrow, pengolahan bahasa alami yang diterapkan untuk pembuatan model Penillaian Esai Pendek Otomatis dibagi menjadi 5 kategori yaitu Leksikal, Morfologikal, Semantik, Sintaktik, dan *Surface* [24]. Kelima kategori tersebut dapat dilihat pada Gambar berikut.



Gambar 2.1. Kategori pemrosesan linguistik

2.2.3 Teks *Pre-Processing*

Teks *pre-processing* adalah salah satu tahap untuk membersihkan data yang berbentuk teks dalam *text mining* [10]. Pembersihan data bertujuan untuk mempersiapkan dokumen berbentuk teks yang tidak terstruktur menjadi data terstruktur untuk diproses lebih lanjut [29]. Belum ada penelitian yang menunjukkan standar penggunakaan tahap teks *pre-processing* pada Penilaian Jawaban Pendek Otomatis. Pada penelitian yang dilakukan oleh Hasanah, umumnya tahap teks *pre-processing* yang digunakan terdiri dari beberapa tahap sebagai berikut [19].

2.2.3.1 *Case Folding*

Case folding merupakan tahap untuk menormalisasi kata dengan mengubah seluruh kata yang memiliki huruf kapital menjadi huruf kecil [10]. Hal tersebut dilakukan karena data mentah dalam bentuk teks yang didapat umumnya tidak terstrukur dan tidak konsisten pada penggunaan huruf kapitalnya. Contohnya, kalimat "*Pre-processing* PaDa Text Mining" akan diubah menjadi "*pre-processing* pada text mining".

Proses ini penting dilakukan pada penelitian ini karena dalam pengoreksian jawaban esai frekuensi penggunaan suatu kata yang menjadi hal utama. Mengabaikan ketidakkonsistenan penggunakan kapital pada kata dalam kalimat menyebabkan kata yang sama akan di anggap berbeda oleh sistem karena satu atau beberapa huruf berbeda dalam penggunaan kapital.

2.2.3.2 *Tokenization*

Tokenization adalah tahap untuk mengubah bentuk kalimat atau paragraf menjadi kata seperti mengubah “pre-processing pada text mining” menjadi “pre-processing”, “pada”, “text”, “mining” [12]. Hasil dari tokenizing ini disebut dengan “token”. Pada tahap ini, token-token tersebut juga akan di filter untuk menghilangkan kata yang tidak terikat dengan kata lain atau tidak relevan kedalam dokumen *stopword* serta menghilangkan karakter tertentu seperti tanda baca [30].

2.2.3.3 *Converting slang word*

Slang word adalah kata yang bersifat tidak baku dan tidak sesuai dengan penulisan tata bahasa [31]. Penggunaan kata *slang* sering kali digunakan pada percakapan sehari-hari. Namun, dalam pendidikan terdapat beberapa kata yang terkadang masih dapat ditoleransi penulisannya seperti kata “yang” yang disingkat menjadi “yg”. Kata tersebut perlu dinormalisasi menjadi kata baku yang sesuai dengan kamus agar dapat terproses lebih lanjut. Proses normalisasi dilakukan dengan menggunakan kamus kata *slang* (*slang dictionary*) terdiri dari 1629 kata untuk Bahasa Indonesia yang didapatkan dari Github taudataid [32].

2.2.3.4 *Lemmatization*

Lemmatization adalah salah satu proses dalam tahap *pre-processing* untuk menormalisasi varian morfologi yang berbeda dengan menghilangkan perfiks dan sufiks pada kata, lalu mengembalikan kata tersebut ke bentuk dasarnya yang memiliki makna [13]. Proses pada *lemmatization* akan menghilangkan imbuhan baik awalan maupun akhiran yang terdapat pada kata lalu mengembalikan kata tersebut ke bentuk dasarnya yang memiliki makna. Hasil *lemmatization* kata “mempertanggungjawabkan” akan menjadi “tanggung jawab”. Contoh lainnya, kata “penerusan” pada *lemmatization* akan berubah menjadi “terus”.

Belum banyak metode *lemmatization* untuk bahasa Indonesia namun terdapat beberapa seperti *Modified Enhanced Confix-Stripping Stemmer* yang di bangun oleh Derwin Suhartono pada tahun 2014 [33]. Pada penelitian yang dilakukan oleh Darwin, akurasi yang didapatkan mencapai 98% namun metode ini masih belum bisa menangani kata yang repetitif dan mengandung imbuhan yang berada di tengah kata

(*infix*). Lalu pada tahun 2018, Namoto membangun kamus morfologi bernama MALINDO *Morph* untuk bahasa Melayu dan Indonesia [34]. MALINDO *Morph* adalah satu-satunya kamus morfologikal yang dapat digunakan untuk menganalisis untuk sebuah token untuk bahasa Melayu dan juga Indonsia. Hasil analisis yang diberikan MALINDO Morph terdiri dari akar kata, *surface form*, prefiks, sufiks, sirkumfiks, dan jenis reduplikasi. Akar kata tersebut yang akan digunakan sebagai hasil *lemmatization* (lemma). Metode ini juga memiliki keunggulan dalam penanganan kata yang bersifat repetitif parsial seperti kata "bebatuan" yang belum bisa dilakukan oleh *Modified Enhanced Conffix Stripping* Stemmer yang dibangun oleh Suhartono.

Selain itu kamus yang digunakan sebagai datanya berasal dari Kamus Dewan (KD) dan Kamus Besar Bahasa Indonesia (KBBI). Untuk token-token yang tidak ada pada kedua kamus tersebut, mereka juga membuat *expanded dictionary* yang merupakan reklasifikasi dari Leipzig Corpora Collection (LCC). *Expanded dictionary* tersebut memiliki enam belas buah berkas subset yang masing-masing subsetnya berisi 300,000 berkas dari LCC. Pada setiap 300,000 berkas tersebut berisi 1,005,007 tipe kata (*case sensitive*) yang tidak ditemukan pada kamus utama (KD dan KBBI). Algoritma ini hanya berfokus untuk afiks produktif lokal dan reduplikasi. Hal tersebut dikarenakan afiks produktif lokal memiliki peran lebih penting dibandingkan dengan non-produktif dan afiks asing seperti "anti-" dan "pre-".

Diketahui inputan adalah *W*, maka algoritma dari MALINDO Morph untuk *lemmatization* pada Bahasa Indonesia terdiri dari beberapa tahap sebagai berikut:

1. Jika *W* adalah karakter non-alfabet, *return W*.
2. Jika *W* ataupun non-kapitalnya sama dengan *w* dalam kata bahasa Inggris, *return W/w*.
3. Jika *W/w* ada dalam kamus, maka *return W/w*.
4. Penggal *clitic* dari *W/w*. Jika hasil penggalan ada dalam kamus, maka *return W/w* pada kamus dan informasi *clitic*nya.
5. Buat set kandidat yaitu $Cand_c$ (*circumfix*), $Cand_p$ (*prefix/proclitic*), dan $Cand_s$ (*suffix*) untuk token *w*.
6. Mencari produk dari $Cand_c \times Cand_p \times Cand_s$ untuk anggota yang elemennya saling sesuai.
7. *Return* hasil dari setiap anggota.

Algoritma pada MALINDO *morph* ini memungkinkan untuk memberikan beberapa *output*. Hal tersebut terjadi karena sebuah kata dapat menghasilkan *term* yang ambigu dalam komposisi morfologinya.

2.2.3.5 *Stemming*

Sama seperti *Lemmatization*, proses *Stemming* pada Bahasa Indonesia juga akan berbeda dengan bahasa lainnya. Metode *stemming* pada bahasa Indonesia hanya akan menghilangkan imbuhan seperti sufiks, prefiks, serta afiks [35]. Penghilangan imbuhan kata pada bahasa Indonesia memungkinkan kata tersebut menjadi tidak bermakna seperti contohnya kata "berupa" menjadi "upa". Hal tersebut dikarenakan "ber-" akan di identifikasi sebagai awalan sehingga menyisakan "upa", sedangkan "upa" ada dalam kamus kata sehingga menghasilkan output "upa"[36]. Selain itu penggunaan *stemming* memungkinkan adanya kegagalan dalam mengembalikan kata menjadi akar katanya [33]. Misalnya, kata "mempertanggungjawabkan" yang di *stemming* akan tetap menjadi "mempertanggungjawabkan". Algoritma *stemming* untuk Bahasa Indonesia yang umumnya diketahui dan digunakan adalah algoritma Porter Stemmer dan algoritma Nazief & Adriani [19].

Berdasarkan penelitian yang dilakukan oleh Agusta, proses *stemming* pada teks menggunakan algoritma Porter memang lebih cepat, namun akurasi yang dihasilkan kecil jika dibandingkan dengan algoritma Nazief dan Andriani [37]. Bahasa Indonesia juga memiliki alat untuk *stemmer* yang berbentuk *library* bernama Sastrawi yang dapat diakses di https://github.com/sastrawi/sastrawi [38]. *Library* ini dibangun berdasarkan penggabungan dari beberapa algoritma seperti *Confix-Stripping* yang dibuat oleh Nazief dan Andriani [39], *Enhanced Confix Stripping Stemmer* oleh Arifin, dan *Modified Enhanced Confix-Stripping Stemmer* yang dibangun Tahitoe untuk memperbaiki masalah *over-stemming* dan *under-stemming* pada algoritma-algoritma sebelumnya [39], [40], [41], [42]. Adapun tahap dari algoritma *stemming* pada Sastrawi adalah sebagai berikut.

1. Pencarian pada kamus, kata yang menjadi inputan akan di cek terlebih dahulu pada kamus kata. Jika kata yang dimaksud ditemukan maka, pencarian akan

berhenti dan mengembalikan kata tersebut sebagai hasil. Kamus yang digunakan pada Sastrawi ini adalah Kateglo yang dapat diakses di http://kateglo.com/.

2. Kata yang memiliki huruf kurang dari tiga misalnya kata “ini”, “dan”, atau “ada” tidak bisa mempunyai afiks, sehingga kata tersebut akan langsung dikembalikan sebagai hasil.
3. Jika kata inputan tidak terdapat di kamus, maka kata tersebut memiliki kemungkinan berimbuhan. Cek terlebih dahulu apakah kata tersebut memiliki kombinasi “be-lah”, “be-an”, “me-i”, “di-i”, “pe-i”, atau “te-i”. Jika ya maka proses dilanjutkan langsung ke tahap nomor 5 yaitu pemenggalan prefiks hingga tahap *Recoding,* lalu kembali ke tahap penghapusan *inflectional suffix*. Selain kata dengan kombinasi tersebut, proses dilakukan dengan tahap yang berurutan. Dalam bahasa Indonesia, terdapat dua tipe sufiks yaitu partikel {‘-lah’, ‘-kah’, ‘-tah’, ‘-pun’} dan kata ganti kepemilikan {‘-ku’, ‘-mu’, ‘-nya’}. Partikel sufiks pada Bahasa Indonesia selalu akan berada pada akhir sebuah kata. Oleh karena itu, penghilangan partikel sufiks akan didahulukan sebelum kata ganti kepemilikan.
4. Penghilangan *derivational suffix*, proses ini akan menghilangkan sufiks turunan seperti {-i, -kan, -an} dari kata yang diberikan. Sufiks ini akan selalu berada di akhir kata sebelum adanya *inflectional suffix*. Maka proses ini akan di jalankan setelah *inflectional suffix* dihilangkan.
5. Penghilangan prefiks turunan, prefiks ini memiliki dua tipe grup yaitu kompleks dan sederhana. Proses pada langkah ini memungkinkan bersifat rekursif karena dalam Bahasa Indonesia, prefiks turunan dapat tertumpuk. Kata dalam Bahasa Indonesia juga memperbolehkan suatu kata mengandung kombinasi dari beberapa prefiks atau kombinasi antara prefiks dan sufiks seperti kata “berkelanjutan”. Namun hanya beberapa kata dengan kombinasi tertentu yang dapat diproses. Kombinasi yang diperbolehkan adalah sebagai berikut:
    a. “di” diikuti awalan tipe “pe-” atau “be-” (Misal. “diberikan” dan “dipertahankan”)
    b. “ke”, diikuti awalan “be-” atau “te-” (Misal. “keterlaluan” dan “kebersamaan”)
    c. “be-”, diikuti awalan “pe-” (Misal. “berpengalaman”)

d. “me-”, diikuti awalan “pe-”, “te-”, atau “be-“ (Misal. “memperbanyak”, “menertawakan”, dan “memberikan”)
e. “pe-”, diikuti awalan “be-” (Misal. “pembelajaran”)

Sedangkan, kata yang memiliki kombinasi awalan dan akhiran yang tidak diperbolehkan untuk diproses adalah “be-i“, “di-an”, “ke-i”, “ke-kan”, “me-an”, “se-i”, “se-kan”, dan “te-an”. Prefiks pada tipe sederhana dapat langsung dihilangkan dari awalan kata karena tidak mentransform kata yang dimaksud seperti prefiks “di-“, “ke-“, dan “se-“. Contohnya kata “sebutir”, “seekor”, atau “sebesar”. Sedangkan, prefiks tipe kompleks seperti “be-“, “me-“, “pe-”, dan “te-“ akan mentransformasi kata yang menggunakan prefiks tersebut seperti kata “pemahaman” yang diubah menjadi “paham”. Untuk prefiks dengan tipe kompleks perlu dilakukan pengecekan dengan aturan pemenggalan sesuai dengan Tabel 2.1

Tabel 2.1 Aturan Pemenggalan *Stemming*

Aturan	**Format kata**	**Pemenggalan**
1	berV...	ber-V...|be-rV...
2	berCAP...	ber-CAP...| dimana C!='r'' dan P!='er'
3	berCAerV...	ber-CaerV... dimana C!=‘‘r‘‘
4	belajar	bel-ajar
5	beC1erC2...	be-C1erC2... dimana C1!={'r'|'l'}
6	terV...	ter-V... | te-rV...
7	terCerV...	ter-CerV... dimana C!='r'
8	terCP...	ter-CP... dimana C!='r' dan P!='er'
9	teC1erC2...	te-C1erC2... dimana C1!='r'
10	me{l|r|w|y}V...	me-{l|r|w|y}V...
11	mem{b|f|v}...	mem-{b|f|v}...
12	mempe...	mem-pe...
13	mem{rV|V}...	me-m{rV|V}... | me-p{rV|V}...
14	men{c|d|j|s|z}...	men-{c|d|j|s|z}...
15	menV...	me-nV... | me-tV
16	meng{g|h|q|k}...	meng-{g|h|q|k}...
17	mengV...	meng-V... | meng-kV...
18	menyV...	me-nyV… | meny-sV

Aturan	Format kata	Pemenggalan
19	mempA...	mem-pA... dengan A!=“e“
20	pe{w|y}V...	pe-{w|y}V...
21	perV...	per-V... | pe-rV...
23	perCAP...	per-CAP... dimana C!=“r“ dan P!=“er“
24	perCAerV...	per-CAerV... dimana C!=“r“
25	pem{b|f|v}...	pem-{b|f|v}...
26	pem{rV|V}...	pe-m{rV|V}... | pe-p{rV|V}...
27	pen{c|d|j|z}...	pen-{c|d|j|z}...
28	penV...	pe-nV... | pe-tV...
29	pengC...	peng-C...
30	pengV...	peng-V... | peng-kV... | (pengV-... jika V=“e“)
31	penyV	peny-sV…
32	pelV...	pe-lV... kecuali “pelajar” yang menghasilkan “ajar
33	peCerV...	per-erV... dimana C!={r|w|y|l|m|n}
34	peCP...	pe-CP... dimana C!={r|w|y|l|m|n} dan P!=“er“
35	terC1erC2...	ter-C1erC2... dimana C1!= „r“
36	peC1erC2...	pe-C1erC2... dimana C1!={r|w|y|l|m|n}
37	CerV…	CerV…| CV...
38	CelV…	CelV… | CV...
39	CemV…	CemV…| CV...
40	CinV…	CinV…| CV...

Keterangan huruf kapital pada tabel:

- C = *Consonant* (huruf mati, semua huruf selain ‘a’,’i’,’u’,’e’,’o’)
- V = *Vowel* (huruf vokal ‘a’,’i’,’u’,’e’,’o’)
- A = *any letter* (huruf apapun)
- P = Partikel

6. *Recoding*.

Proses *recoding* dilakukan ketika kemungkinan pertama pada tahap 5 menghasilkan kata yang tidak terdapat didalam kamus. Sehingga, pada tahap ini dilakukan untuk mendapatkan kemungkinan lain dari aturan pemenggalan pada Tabel

2.1 jika ada. Proses ini merupakan proses untuk melakukan percobaan dengan menghapus atau menggantkan karakter pada huruf pertama pada kata setelah menghilangkan prefiks yang dipenggal sesuai dengan aturan pemenggalan.

Jika semua tahap telah dilakukan namun kata tersebut tidak terdapat dalam kamus, maka kata tersebut diasumsikan sebagai akar katanya. Ketika sebuah kata berhasil melewati semua tahap secara benar, maka proses *stemming* dianggap berhasil. Namun, terdapat beberapa kasus dimana *stem* gagal dikategorikan dikarenakan diluar dari batasan seperti sebagai berikut [33]:

1. Kata asing, teks yang diproses mengandung kata selain Bahasa Indonesia, sehingga kata tersebut tidak tersimpan dalam kamus.
2. Imbuhan dalam kata, imbuhan tersebut disisipkan kedalam sebuah kata sehingga kata yang dimaksud dikenali sebagai *stem* itu sendiri.
3. Bahasa tidak baku, penggunaan bahasa tidak baku dalam teks tidak akan dikenali karena tidak terdapat di dalam kamus.

Selain itu, terdapat juga jenis-jenis *error* dari *stemming* yaitu sebagai berikut:

1. *Overstemming*, penghilangan imbuhan yang berlebihan. Contohnya, kata “penyidikan” akan menghasilkan “sidi”.

2. *Understemming*, merupakan kebalikan dari *overstemming*, yaitu penghilangan imbuhan yang terlalu sedikit yang tetap menyebabkan ketidaksesuaian *stem*. Contohnya, kata “mengalami” menjadi “alami”.

2.2.4 *Term Frequency-Inverse Document Frequency* (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) adalah sebuah metode untuk pembobotan kata dengan tujuan untuk memberikan nilai pada *term* yang ada pada suatu dokumen [43]. Metode ini merupakan gabungan dari metode TF (*Term Frequency*) dan IDF (*Inverse Document Frequency*) yang diusung oleh Spark Jones agar dapat memberikan performa yang lebih baik dalam memperbaiki nilai *recall* dan *precision* [43]. Pada metode ini, bobot setiap kata dipetakan ke dalam lingkungan vektor sedemikan rupa hingga membentuk lingkungan vektor berdimensi n [44]. Nilai TF-IDF didapatkan dari mengalikan hasil TF dengan IDF dengan persamaan yang ditunjukkan pada Rumus 2.1. [45].

$$w_{t,d} = tf_{t,d} \times idf_t \tag{2.1}$$

Keterangan Rumus 2.1:

$w_{t,d}$ = Bobot TF-IDF

$tf_{t,d}$ = Frekuensi dari suatu kata yang muncul pada suatu dokumen

idf_t = Jumlah kata yang muncul pada seluruh dokumen

Term Frequency (TF) adalah sebuah matrik untuk menghitung seberapa sering frekuensi dari suatu kata yang muncul pada suatu dokumen. Persamaan untuk TF ditunjukkan pada Rumus 2.2.

$$tf_{t,d} = \frac{N_{t,d}}{N_d} \tag{2.2}$$

Keterangan Rumus 2.2:

$N_{t,d}$ = Jumlah kemunculan *term* (t) pada satu dokumen (d)

N_d = Jumlah kata pada dokumen

Sedangkan IDF adalah jumlah kata yang muncul pada seluruh dokumen. Persamaan untuk IDF ditunjukkan pada Rumus 2.3.

$$idf_t = log \frac{\text{n}}{(n_k + 1)} \quad (2.3)$$

Keterangan Rumus 2.3:

n = Jumlah dokumen

n_k = Jumlah dokumen yang memiliki term yang dimaksud.

2.2.5 *Cosine Similarity*

Cosine Similarity adalah metode untuk mengukur kesamaan dalam *retrieval information* dan ukuran dari sisi pandang antar dua dokumen. Setiap vektor di representasikan dalam tiap kata pada dokumen berbentuk teks yang berasal dari komparasi dalam bentuk segitiga sehingga hukum *cosine* dapat implementasikan menjadi [46]:

$$Cosine\ Similarity = \cos(\theta) = \frac{A.B}{\left||A|\right|.||B||}$$

$$Cosine\ Similarity(q,d) = \frac{\sum_{k=1}^{t} w_{qk} \times w_{dk}}{\sqrt{\sum_{k=1}^{t}(w_{qk})^2} \cdot \sqrt{\sum_{k=1}^{t}(w_{dk})^2}} \quad (2.4)$$

Keterangan Rumus 2.4:

w_{qk} = nilai dari *query*$_k$

w_{dk} = nilai dari dokumen$_k$

Cosine Similarity digunakan untuk membandingkan level similaritas dari dokumen dengan konsep derajat cosine dimana hasilnya merupakan rentang antara 0 dan 1 [46]. Semakin dekat angka dengan 0 menunjukkan bahwa semakin rendah tingkat kesamaan kedua dokumen tersebut. Sebaliknya, semakin dekat angka dengan 1 maka semakin tinggi tingkat kesamaan kedua dokumen tersebut.

2.2.6 *Mean Absolute Error* (MAE)

Mean Absolute Error adalah salah satu jenis model evaluasi yang digunakan bersama dengan model regresi. Berdasarkan penelitian Willmot, metode ini cocok digunakan karena memberikan hasil pengukuran kesalahan yang lebih natural dan tidak ambigu [47]. Hal tersebut dikarenakan metode ini tidak sensitif terhadap *outlier* melainkan hanya benar-benar menghitung nilai kesalahannya dari setiap individu saja.

Hasil dari evaluasi ini adalah rata-rata dari nilai mutlak dari kesalahan pada setiap prediksi individual. Nilai kesalahan pada prediksi merupakan nilai perbedaan antara nilai sebenarnya dengan nilai yang diprediksi [48].

$$MAE = \frac{\sum_{i=1}^{n} |(X_i - Y_i)|}{n} \quad (2.5)$$

Keterangan Rumus 2.5:

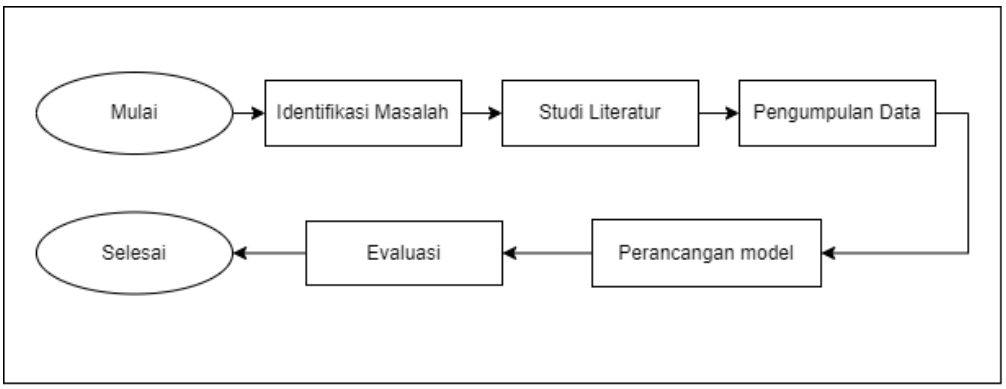
X_i = nilai prediksi

Y_i = nilai sesungguhnya

BAB III
METODE PENELITIAN

3.1 Alur Penelitian

Alur pada penelitian ini dapat dilihat pada Gambar 3.1. Penelitian dimulai dengan melakukan identifikasi masalah yang ditemukan, lalu melakukan studi literatur penelitian terkait hingga melakukan evaluasi terhadap hasil Penilaian Jawaban Pendek Otomatis menggunakan penerapan metode *stemming* dan *lemmatization*.



Gambar 3.1 *Flowchart* Alur Penelitian

3.2 Penjabaran Langkah Penelitian

Berikut adalah penjabaran dari tiap langkah pada alur penelitian yang telah digambarkan pada Gambar 3.1.

3.2.1 Studi Literatur

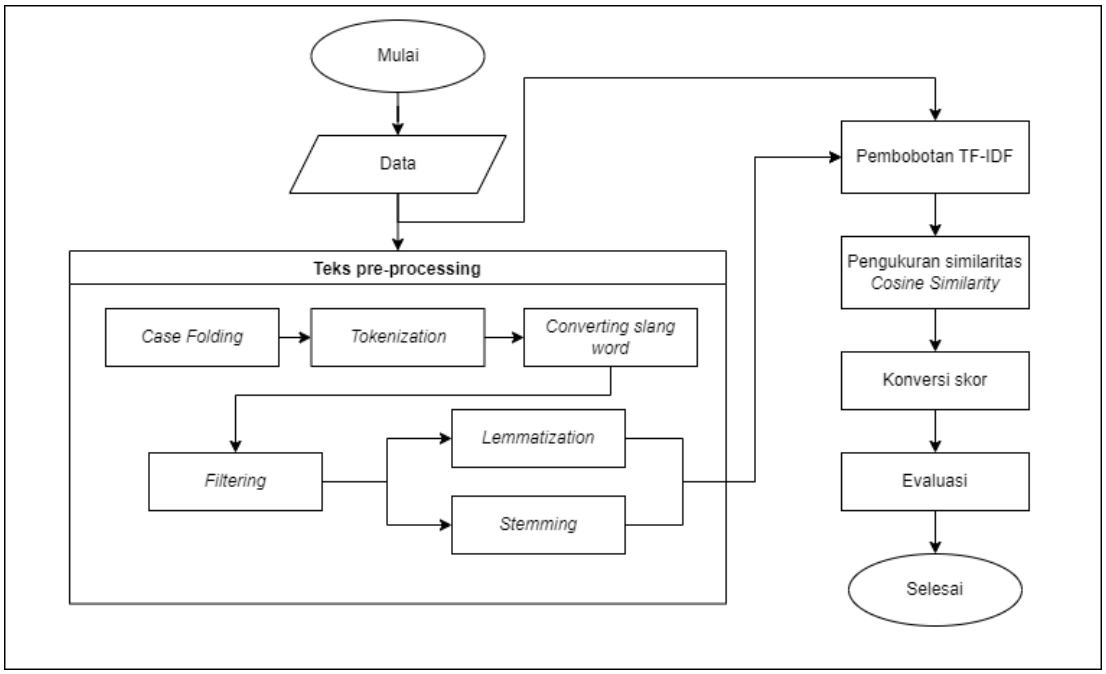
Pada tahap studi literatur, peneliti melakukan peninjauan terkait topik penilaian jawaban pendek otomatis, *stemming*¸dan *lemmatization* dari sumber-sumber seperti buku, jurnal, dan sumber lainnya

3.2.2 Pengumpulan Data

Tahap pengumpulan data dilakukan untuk mendapatkan data yang akan digunakan untuk dapat diolah dalam model Penilaian Jawaban Pendek Otomatis. Detail data yang digunakan pada penelitian ini tertulis pada subbab 3.3.2.

3.2.3 Pengembangan Model

Tahap ini adalah tahap untuk merancang pemodelan Penilaian Jawaban Pendek Otomatis. Adapun alur dari pengembangan model Penilaian Jawaban Pendek Otomatis ditunjukkan pada Gambar 3.2.



Gambar 3.2. Alur Pemodelan Penilaian Jawaban Pendek Otomatis

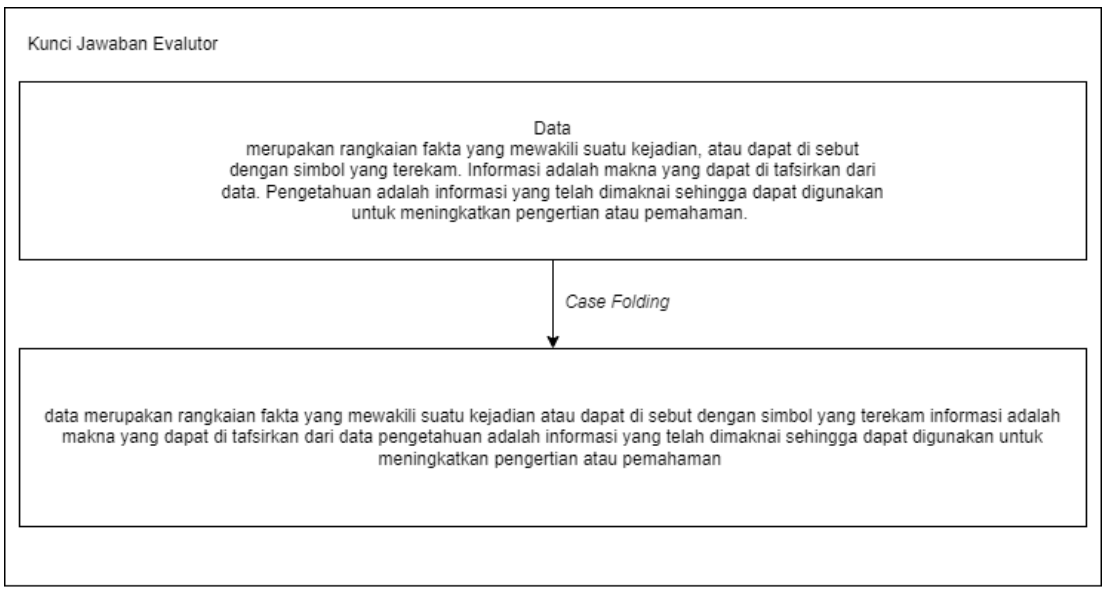
Pada tahap pertama, data akan diolah pada tahap teks *pre-processing.* Data teks yang diproses dalam tahap teks *pre-processing* adalah data jawaban dari siswa dan jawaban dari evaluator (kunci jawaban). Misalkan pertanyaan yang diberikan adalah "Apa perbedaan dari data, informasi dan pengetahuan?", maka terdapat satu buah data kunci jawaban dan beberapa jawaban siswa yang perlu diproses. Berikut adalah contoh dari data kunci jawaban dan jawaban siswa yang akan diproses dalam tahap teks *pre-processing*.

Kunci Jawaban Evalutor : “Data merupakan rangkaian fakta yang mewakili suatu kejadian, atau dapat di sebut dengan simbol yang terekam. Informasi adalah makna yang dapat di tafsirkan dari data. Pengetahuan adalah informasi yang telah dimaknai sehingga dapat digunakan untuk meningkatkan pengertian atau pemahaman.”

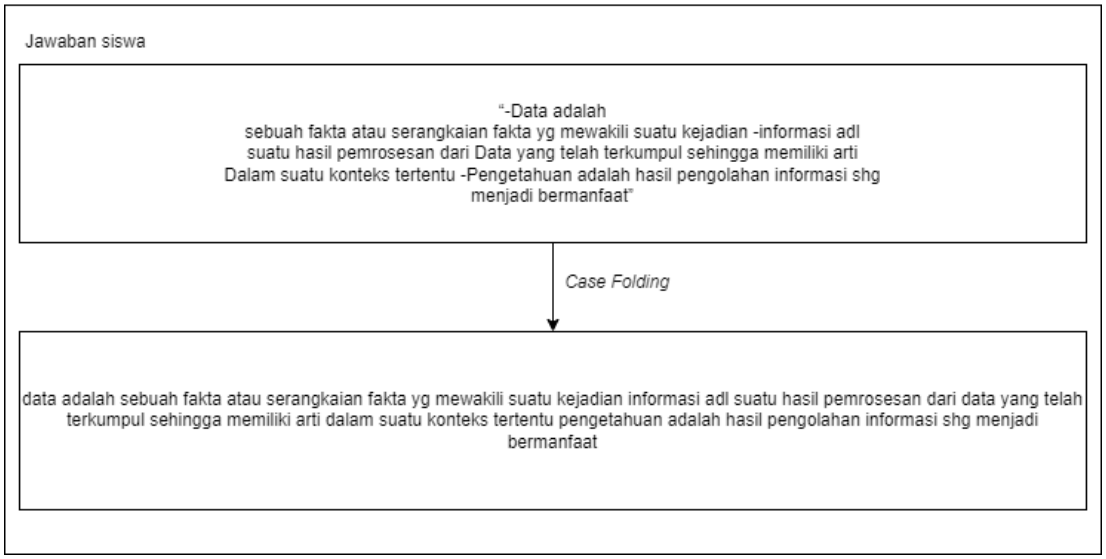
Jawaban Siswa: “-Data adalah sebuah fakta atau serangkaian fakta yg mewakili suatu kejadian -informasi adl suatu hasil pemrosesan dari Data yang telah terkumpul sehingga memiliki arti Dalam suatu konteks tertentu -Pengetahuan adalah hasil pengolahan informasi shg menjadi bermanfaat”

1. *Case Folding*

Pada tahap *pre-processing* pertama yaitu *case folding*, semua huruf dalam dokumen diubah menjadi huruf kecil serta tanda baca, angka, dan seluruh karakter selain huruf akan dihilangkan. Hasil dari tahap *case folding* ditunjukkan pada Gambar 3.3.



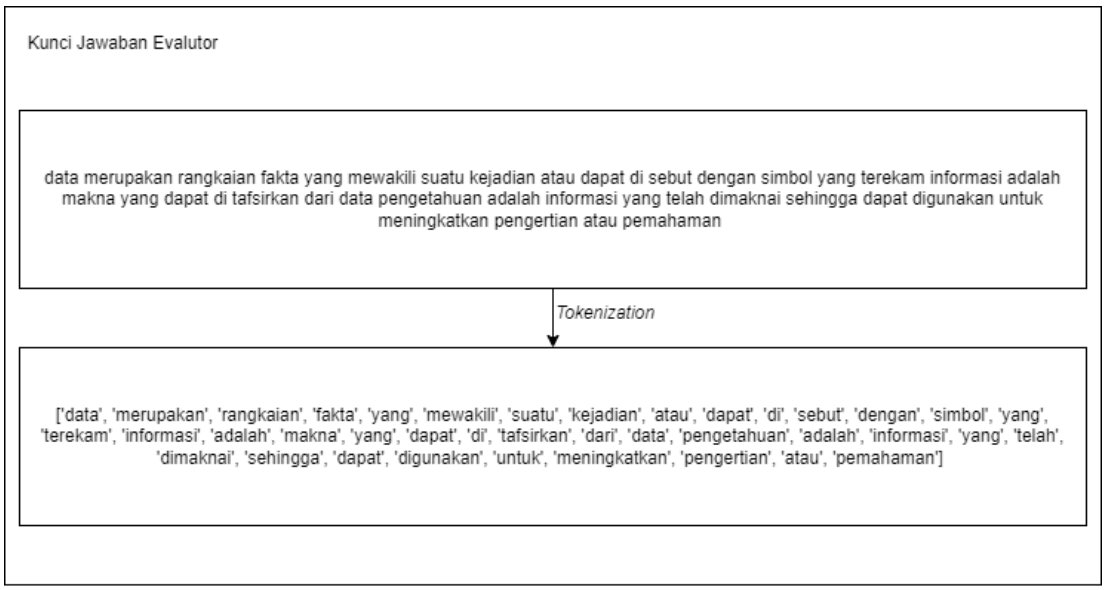
Gambar 3.3 Hasil *case folding* kunci jawaban



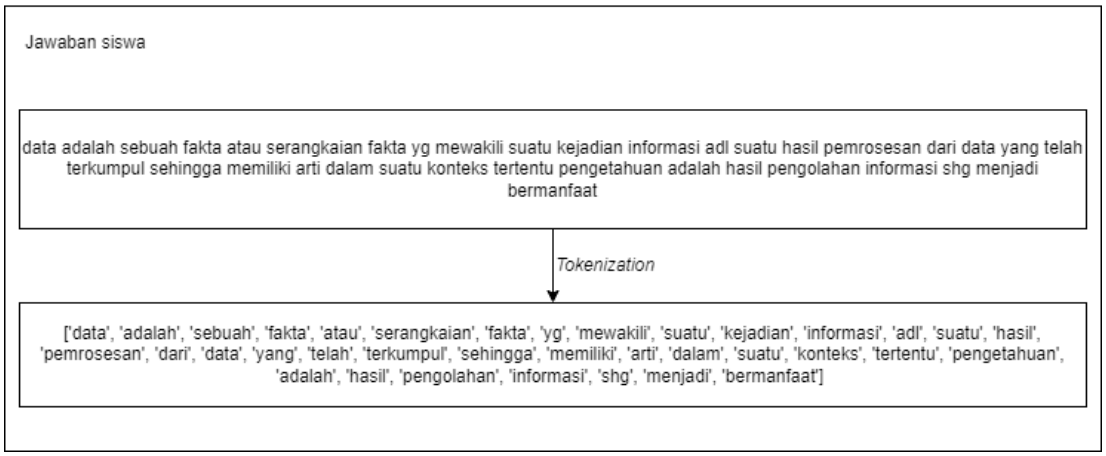
Gambar 3.4 Hasil *case folding* jawaban siswa

2. *Tokenization*

Tahap selanjutnya adalah teks hasil dari tahap *case folding* akan dipotong menjadi kumpulan kata. Kata akan dipisah dengan *delimiter* berupa spasi (" ").



Gambar 3.5 Hasil *tokenization* kunci jawaban

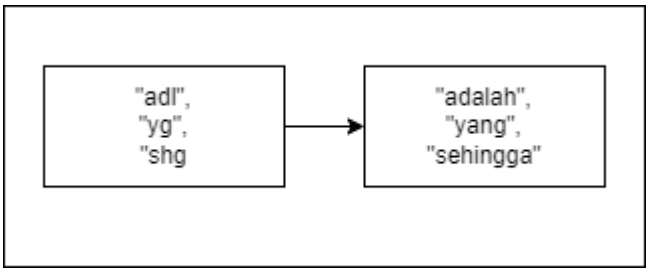


Gambar 3.6 Hasil *tokenization* jawaban siswa

3. *Converting slang word*

Data teks hasil dari tahap *tokenization* selanjutnya akan dilakukan pengecekan apakah didalam data tersebut terdapat kata yang tidak sesuai dengan standar penulisan Bahasa Indonesia sesuai dengan Kamus Besar Bahasa Indonesia. Proses pengecekan dilakukan dengan membandingan data inputan (jawaban dan kunci jawaban) dengan kamus *slang* untuk Bahasa Indonesia.

Terdapat beberapa kata *slang* yang terdapat pada jawaban siswa seperti kata “adl”, “yg”, dan “shg”. Kata-kata tersebut akan di konversi menjadi kata yang sesuai dengan kaidah penulisan yang benar. Berikut adalah contoh kata *slang* sebelum dan sesudah di konversi ditunjukkan pada Gambar 3.7.

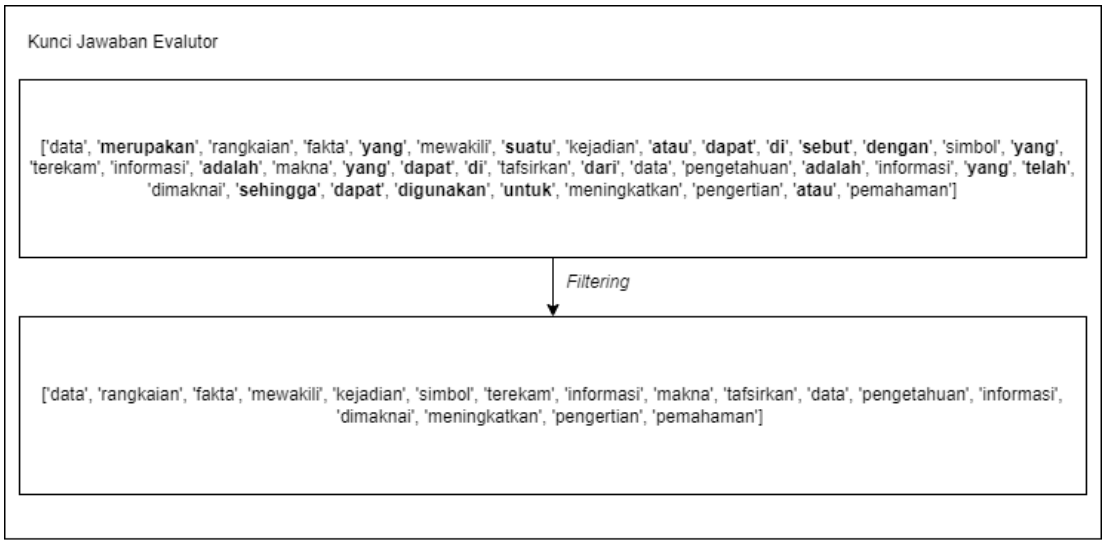


Gambar 3.7 Hasil *converting slang word*

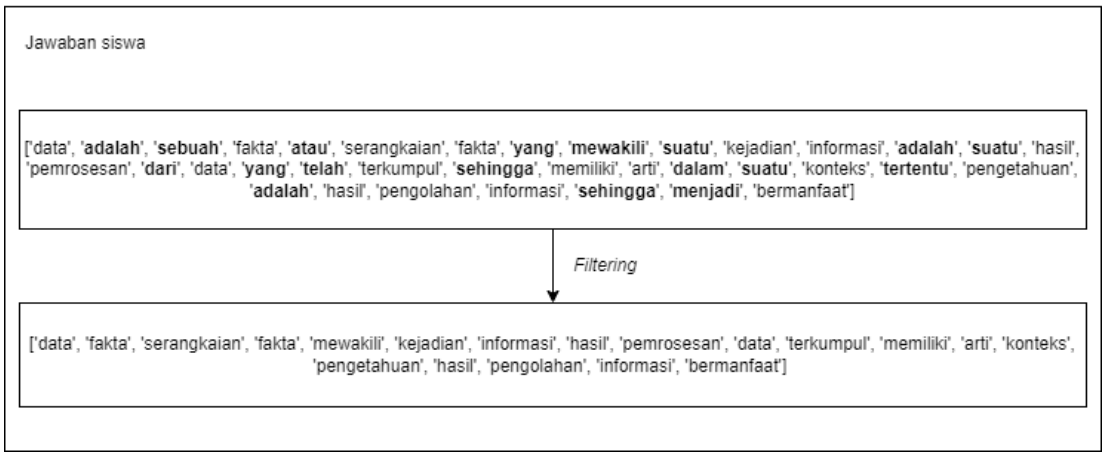
4. *Filtering*

Pada tahap *filtering*, *stopword* atau kata yang tidak memiliki makna atau tidak *berarti* dan tidak relevan akan dibuang. Pemilihan kata yang akan dibuang berdasarkan

kamus *stopword* untuk bahasa Indonesia. Jika terdapat kata yang ada dalam *stopword* pada teks, maka kata tersebut akan dibuang.



Gambar 3.8 Hasil *filtering* kunci jawaban



Gambar 3.9 Hasil *filtering* jawaban siswa

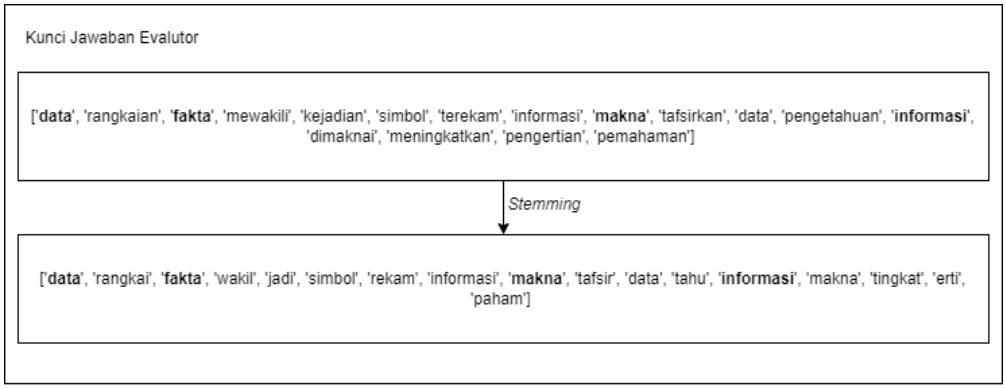
5. *Stemming*

Stemming akan menghilangkan imbuhan seperti sufiks, prefiks, serta afiks untuk mengembalikan suatu kata ke bentuk akarnya. Pada proses ini, metode *stemming* yang digunakan adalah dengan menggunakan *library* Sastrawi. Simulasi proses *stemming* menggunakan algoritma Nazief dan Adriani adalah sebagai berikut.

7. Pencarian pada kamus, kata yang menjadi inputan akan di cek terlebih dahulu pada kamus kata. Jika kata yang dimaksud ditemukan maka, pencarian akan berhenti dan mengembalikan kata tersebut sebagai hasil. Kamus yang

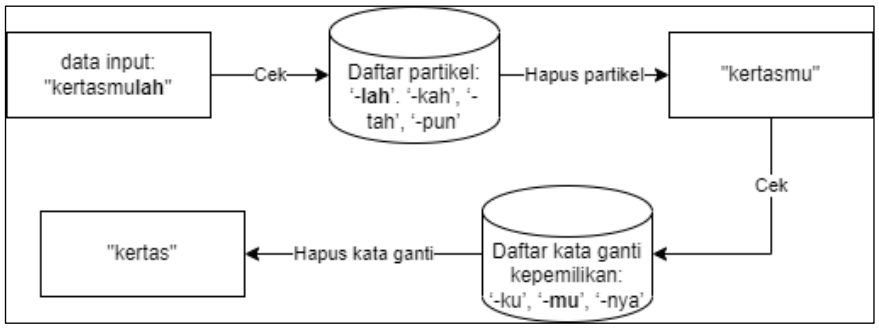
digunakan pada Sastrawi ini adalah Kateglo yang dapat diakses di http://kateglo.com/.

Berdasarkan gambar 3.10, kata yang dicetak tebal seperti “data”, “fakta”, “makna”, dan “informasi” sudah merupakan kata dasar, sehingga kata-kata tersebut tidak akan diproses lebih lanjut dan akan dikembalikan sebagai hasil.



Gambar 3.10 Contoh kata yang sudah merupakan kata dasar

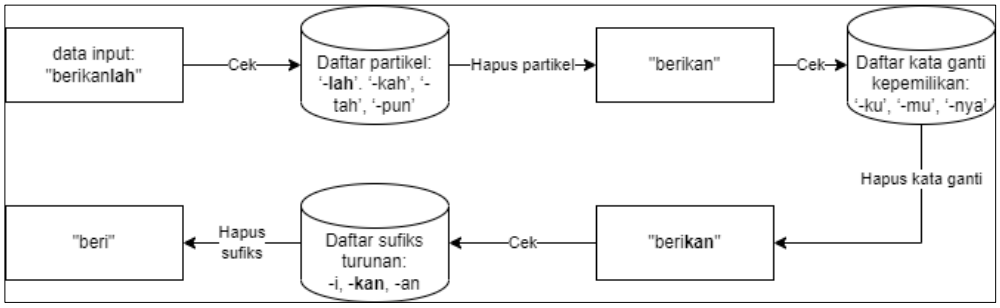
8. Kata yang memiliki huruf kurang dari tiga misalnya kata “ini”, “dan”, atau “ada” tidak bisa mempunyai afiks, sehingga kata tersebut akan langsung dikembalikan sebagai hasil.
9. Jika kata inputan tidak terdapat di kamus, maka kata tersebut memiliki kemungkinan berimbuhan. Cek terlebih dahulu apakah kata tersebut memiliki kombinasi “be-lah”, “be-an”, “me-i”, “di-i”, “pe-i”, atau “te-i”. Jika ya maka proses dilanjutkan langsung ke tahap nomor 5 yaitu pemenggalan prefiks hingga tahap *Recoding*, lalu kembali ke tahap penghapusan *inflectional suffix*. Selain kata dengan kombinasi tersebut, proses dilakukan dengan tahap yang berurutan. Dalam bahasa Indonesia, terdapat dua tipe sufiks yaitu partikel {‘-lah’, ‘-kah’, ‘-tah’, ‘-pun’} dan kata ganti kepemilikan {‘-ku’, ‘-mu’, ‘-nya’}. Partikel sufiks pada Bahasa Indonesia selalu akan berada pada akhir sebuah kata. Oleh karena itu, penghilangan partikel sufiks akan didahulukan sebelum kata ganti kepemilikan.



Gambar 3.11 Proses penghapusan *inflectional suffix*

Pada Gambar 3.11, dimisalkan kata "kertasmulah" akan dihilangkan partikelnya menjadi "kertasmu". Setelah itu, kata ganti kepemilikan "-mu" akan dihilangkan menjadi "kertas".

10. Penghilangan *derivational suffix*, proses ini akan menghilangkan sufiks turunan seperti {-i, -kan, -an} dari kata yang diberikan. Sufiks ini akan selalu berada di akhir kata sebelum adanya *inflectional suffix*. Maka proses ini akan di jalankan setelah *inflectional suffix* dihilangkan.



Gambar 3.12 Proses penghapusan *derivational suffix*

Pada Gambar 3.12 diberikan contoh kata "berikanlah". Kata tersebut akan dihilangkan partikelnya menjadi "berikan". Setelah itu, dilakukan pengecekan apakah kata tersebut memiliki kata ganti kepemilikan. Apabila kata tersebut memiliki kata ganti kepemilikian, maka kata ganti tersebut akan dihilangkan terlebih dahulu. Karena kata "berikan" tidak memiliki kata ganti kepemilikan, maka proses penghilangan *derivational suffix* akan langsung dilakukan dengan menghilangkan "-kan" pada kata. Sehingga kata "berikan" memiliki hasil akhir "beri" pada proses ini.

11. Penghilangan prefiks turunan, prefiks ini memiliki dua tipe grup yaitu kompleks dan sederhana. Proses pada langkah ini memungkinkan bersifat rekursif karena dalam Bahasa Indonesia, prefiks turunan dapat tertumpuk. Kata dalam Bahasa Indonesia juga memperbolehkan suatu kata mengandung kombinasi dari beberapa prefiks atau kombinasi antara prefiks dan sufiks seperti kata "berkelanjutan". Namun hanya beberapa kata dengan kombinasi tertentu yang dapat diproses. Kombinasi yang diperbolehkan adalah sebagai berikut:
    f. "di" diikuti awalan tipe "pe-" atau "be-" (Misal. "diberikan" dan "dipertahankan")
    g. "ke", diikuti awalan "be-" atau "te-" (Misal. "keterlaluan" dan "kebersamaan")
    h. "be-", diikuti awalan "pe-" (Misal. "berpengalaman")
    i. "me-", diikuti awalan "pe-", "te-", atau "be-" (Misal. "memperbanyak", "menertawakan", dan "memberikan")
    j. "pe-", diikuti awalan "be-" (Misal. "pembelajaran")

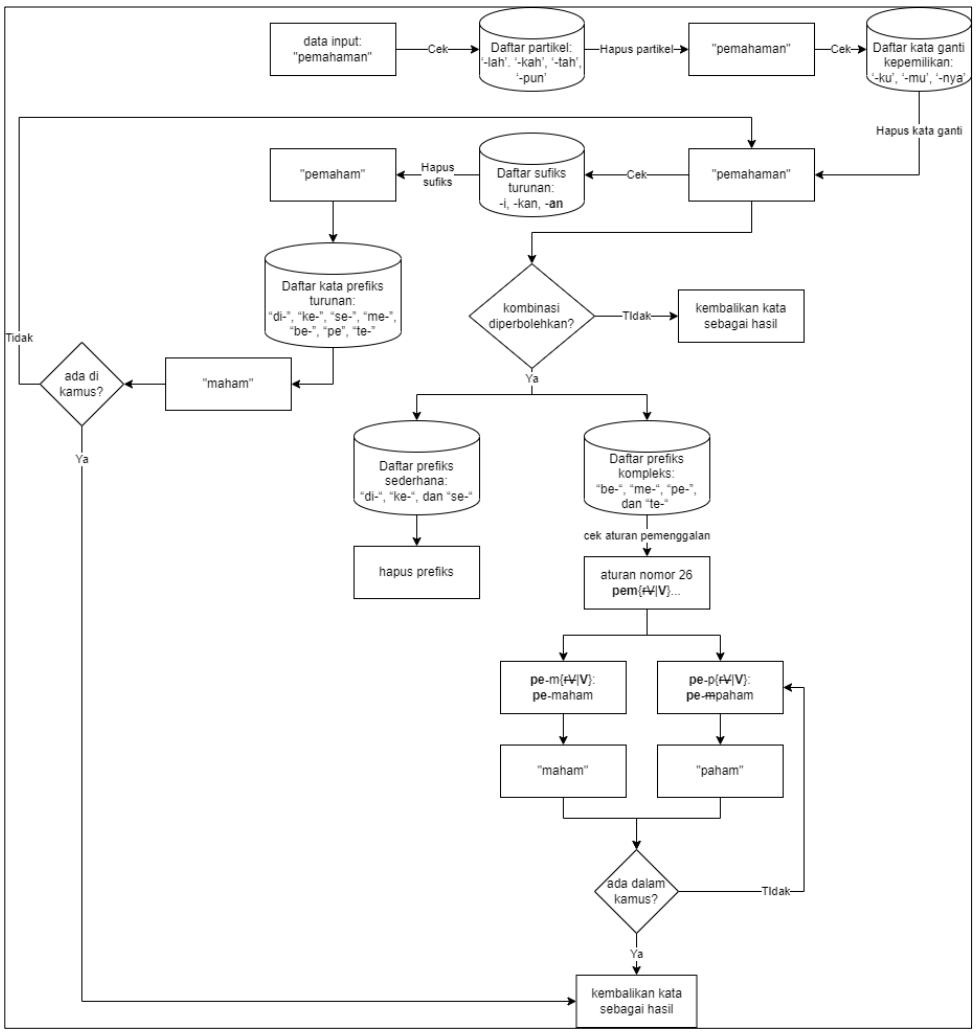
    Sedangkan, kata yang memiliki kombinasi awalan dan akhiran yang tidak diperbolehkan untuk diproses adalah "be-i", "di-an", "ke-i", "ke-kan", "me-an", "se-i", "se-kan", dan "te-an".

    Prefiks pada tipe sederhana dapat langsung dihilangkan dari awalan kata karena tidak mentransform kata yang dimaksud seperti prefiks "di-", "ke-", dan "se-". Contohnya kata "sebutir", "seekor", atau "sebesar". Sedangkan, prefiks tipe kompleks seperti "be-", "me-", "pe-", dan "te-" akan mentransformasi kata yang menggunakan prefiks tersebut seperti kata "pemahaman" yang diubah menjadi "paham".

    Kata "pemahaman" setelah melalui tahap penghilangan *derivational suffix* akan menjadi "pemaham". Sufiks "pe-" pada kata "pemaham" akan dicoba dihilangkan terlebih dahulu sehingga menghasilkan kata "maham". Namun, karena kata "maham" tidak terdapat dalam kamus, maka akan terjadi iterasi untuk mengecek apakah kata sebelum dilakukan penghilangan *derivational suffix* merupakan kombinasi yang dilarang. Jika tidak, maka lakukan pengecekan untuk tipe prefiks yang terdapat pada kata. Karena prefiks pada

kata “pemahaman” adalah “pe-“ yang masuk ke tipe kompleks, maka perlu adanya pengecekan aturan pemenggalan sesuai dengan Tabel. 2.1. Diketahui kata pemahaman memenuhi aturan nomor 26 yaitu “pem{rv|V}” sehingga hasil pemenggalan memiliki 2 pilihan yaitu “pe-m{rV|V}” atau “pe-p{rV|V}”. Pemenggalan dilakukan dengan menghilangkan karakter “-“ dan seluruh karakter sebelum tanda “-“ tersebut. Hasil dari pemenggalan “pe-m{rV|V}” adalah “maham”, yang mana tidak ditemukan pada kamus. Sehingga terjadilah proses *Recoding* untuk mendapatkan kemungkinan kedua yaitu hasil pemenggalan dengan aturan “pe-p{rV|V}”. Aturan pemenggalan tersebut menghasilkan kata “paham”, sehingga hasil *stemming* dari kata pemahaman adalah “paham”

Proses *stemming* kata “pemahaman” diilustrasikan pada Gambar 3.13.



Gambar 3.13 Proses pemenggalan prefiks

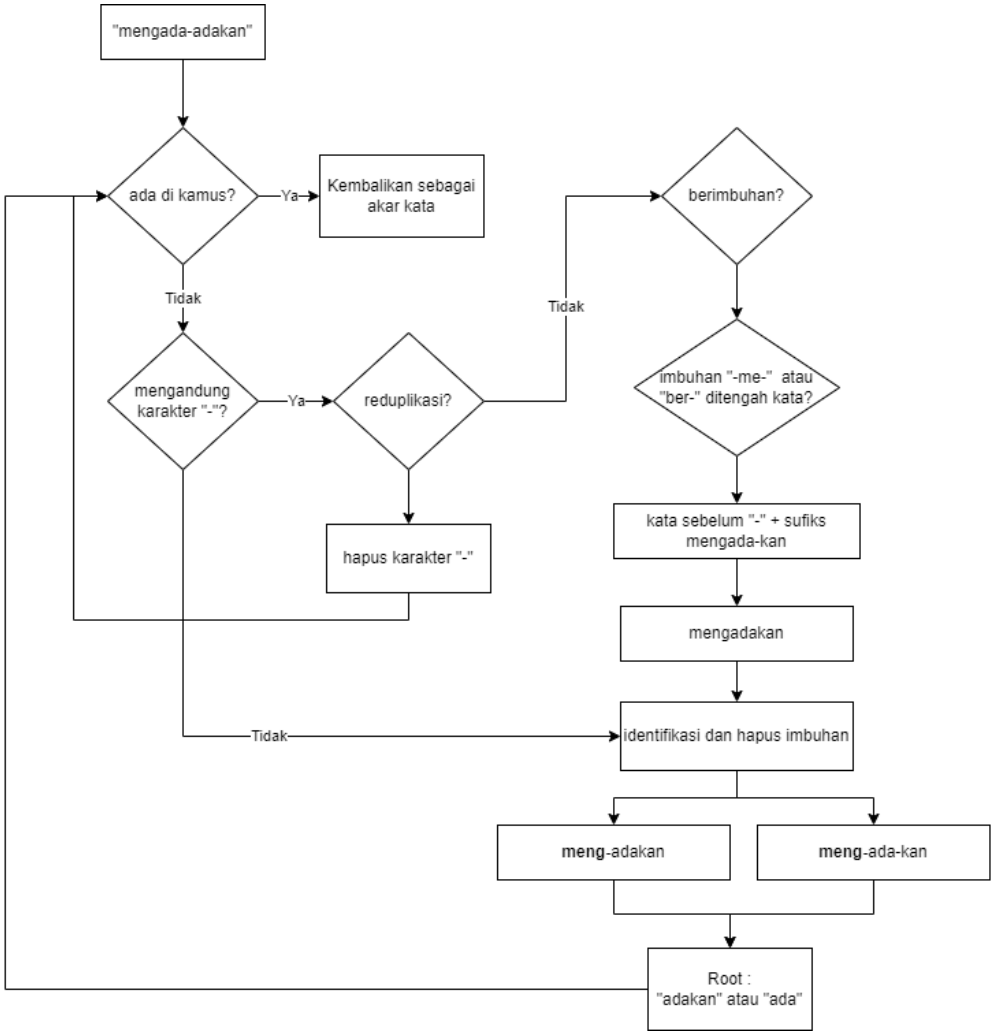
12. *Recoding*.

Proses *recoding* dilakukan ketika kemungkinan pertama pada tahap 5 menghasilkan kata yang tidak terdapat didalam kamus. Sehingga, pada tahap ini dilakukan untuk mendapatkan kemungkinan lain dari aturan pemenggalan pada Tabel 2.1 jika ada. Proses ini merupakan proses untuk melakukan percobaan dengan menghapus atau menggantkan karakter pada huruf pertama pada kata setelah menghilangkan prefiks yang dipenggal sesuai dengan aturan pemenggalan.

6. *Lemmatization*

Lemmatization akan mengembalikan suatu kata ke bentuk dasarnya dengan menghilangkan imbuhan baik awalan maupun akhiran serta mengembalikan makna dari kata tertentu. Algoritma yang digunakan untuk *lemmatization* pada penelitian ini adalah MALINDO *Morph*. Misalkan kata inputan yang akan diproses adalah “mengada-adakan”, maka simulasi proses *lemmatization* diilustrasikan pada Gambar 3.14.

1. Jika inputan adalah karakter non-alfabet, kembalikan inputan sebagai hasil. Kata “mengada-adakan” adalah karakter alfabet, maka proses dilanjutkan ke tahap selanjutnya.
2. Jika inputan ataupun non-kapitalnya sama dengan kata dalam bahasa Inggris, kembalikan inputan sebagai hasil. Kata “mengada-adakan” adalah kata dalam bahasa Indonesia, maka proses dilanjutkan.
3. Jika *W/w* ada dalam kamus, maka *return W/w*. Kata “mengada-adakan” tidak ada dalam kamus, maka proses dilanjutkan.
4. Penggal *clitic* dari *W/w*. Jika hasil penggalan ada dalam kamus, maka *return W/w* pada kamus dan informasi *clitic*nya.
5. *Return* hasil dari setiap anggota.



Gambar 3.13 Proses *lemmatization*

3.2.3.1 Pembobotan Term Frequency – Inverse Document Frequency (TF-IDF)

Untuk membuktikan benar bahwa tahap *pre-processing* berpengaruh terhadap akurasi model, data yang akan dibobotkan menggunakan TF-IDF dibagi menjadi dua yaitu data hasil *pre-processing* dan data mentah atau tanpa melalui tahap *pre-processing.* Misalkan data yang akan dibobotkan adalah seperti pada Tabel 3.2

Tabel 3.2. Contoh teks yang akan dibobotkan

Dokumen	***Term***	**Jumlah kata (Nd)**
Query (Kunci Jawaban)	['mouse', 'perangkat', 'keras', 'komputer', 'fungsi', 'kendali', 'kursor', 'tampil', 'gui', 'graphical', 'user', 'interface', 'masuk', 'input', 'gera', 'kursor', 'tekan',	37

	'tombol', 'click', 'menaikturunkan', 'tampil', 'layar', 'scroll', 'mouse', 'kabel', 'wireless', 'bahasa', 'indonesia', 'mouse', 'arti', 'tetikus', 'bentuk', 'mouse', 'rupa', 'tikus', 'badan', 'gembung']	
D1 (Jawaban Siswa)	['mouse', 'komputer', 'perangkat', 'input', 'fungsi', 'kontrol', 'kursor', 'gui', 'antarmuka', 'guna', 'grafis', 'arah', 'gerak', 'naikturun', 'layar', 'pindah', 'pilih', 'teks', 'ikon', 'file', 'folder', 'layar', 'monitor']	23

Sebelum melakukan pembobotan, hal pertama yang perlu dilakukan adalah membuat kamus kata. Kamus merupakan kumpulan kata berbeda dari seluruh dokumen total dokumen yang ada.

Perhitungan bobot dimulai dengan menghitung *term* pada kunci jawaban dengan jawaban siswa. Jika terdapat *term* yang dimaksud pada dokumen jawaban siswa, maka nilai *term* tersebut adalah 1. Setelah mendapatkan jumlah *term* pada masing-masing dokumen, maka tahap selanjutnya adalah menghitung jumlah kata dari tiap dokumen (Nd). Setelah mendapatkan hasil Nd, maka TF sudah dapat dihitung dengan membagi *term* pada masing-masing dokumen dengan Nd.

Tabel 3.1. Hasil perhitungan TF dan IDF

Term	**Term (Nt,d)**		**df**	**Jumlah kata (Nd)**		**N**	**TF (Nt,d/df)**		**IDF (LOG(N/Nt,d))**
	q	**d1**		**Nq1**	**Nd1**		**q**	**d1**	
guna	0	1	1	37	23	2	0,000	0,043	0,000
kontrol	0	1	1	37	23	2	0,000	0,043	0,000
tekan	1	0	1	37	23	2	0,027	0,000	0,000
tampil	2	0	2	37	23	2	0,054	0,000	-0,176
bahasa	1	0	1	37	23	2	0,027	0,000	0,000
pilih	0	1	1	37	23	2	0,000	0,043	0,000
gembung	1	0	1	37	23	2	0,027	0,000	0,000
scroll	1	0	1	37	23	2	0,027	0,000	0,000
tetikus	1	0	1	37	23	2	0,027	0,000	0,000
ikon	0	1	1	37	23	2	0,000	0,043	0,000
user	1	0	1	37	23	2	0,027	0,000	0,000

Term	Term (Nt,d)		df	Jumlah kata (Nd)		N	TF (Nt,d/df)		IDF (LOG(N/Nt,d))
	q	d1		Nq1	Nd1		q	d1	
gerak	0	1	1	37	23	2	0,000	0,043	0,000
arti	1	0	1	37	23	2	0,027	0,000	0,000
folder	0	1	1	37	23	2	0,000	0,043	0,000
teks	0	1	1	37	23	2	0,000	0,043	0,000
perangkat	1	1	2	37	23	2	0,027	0,043	-0,176
monitor	0	1	1	37	23	2	0,000	0,043	0,000
kursor	2	1	2	37	23	2	0,054	0,043	-0,176
bentuk	1	0	1	37	23	2	0,027	0,000	0,000
wireless	1	0	1	37	23	2	0,027	0,000	0,000
arah	0	1	1	37	23	2	0,000	0,043	0,000
antarmuka	0	1	1	37	23	2	0,000	0,043	0,000
grafis	0	1	1	37	23	2	0,000	0,043	0,000
rupa	1	0	1	37	23	2	0,027	0,000	0,000
fungsi	1	1	2	37	23	2	0,027	0,043	-0,176
input	1	1	2	37	23	2	0,027	0,043	-0,176
gera	1	0	1	37	23	2	0,027	0,000	0,000
indonesia	1	0	1	37	23	2	0,027	0,000	0,000
click	1	0	1	37	23	2	0,027	0,000	0,000
file	0	1	1	37	23	2	0,000	0,043	0,000
interface	1	0	1	37	23	2	0,027	0,000	0,000
kendali	1	0	1	37	23	2	0,027	0,000	0,000
menaiktu-runkan	1	0	1	37	23	2	0,027	0,000	0,000
mouse	4	1	2	37	23	2	0,108	0,043	-0,176
tombol	1	0	1	37	23	2	0,027	0,000	0,000
kabel	1	0	1	37	23	2	0,027	0,000	0,000
komputer	1	1	2	37	23	2	0,027	0,043	-0,176
layar	1	2	2	37	23	2	0,027	0,087	-0,176
keras	1	0	1	37	23	2	0,027	0,000	0,000
pindah	0	1	1	37	23	2	0,000	0,043	0,000
graphical	1	0	1	37	23	2	0,027	0,000	0,000
tikus	1	0	1	37	23	2	0,027	0,000	0,000
badan	1	0	1	37	23	2	0,027	0,000	0,000
masuk	1	0	1	37	23	2	0,027	0,000	0,000
naikturun	0	1	1	37	23	2	0,000	0,043	0,000

Term	Term (Nt,d)		df	Jumlah kata (Nd)		N	TF (Nt,d/df)		IDF (LOG(N/Nt,d))
	q	d1		Nq1	Nd1		q	d1	
gui	1	1	2	37	23	2	0,027	0,043	-0,176

Setelah mendapatkan TF, selanjutnya adalah menghitung IDF menggunakan persamaan 2.2. Nilai IDF_t adalah nilai IDF dari setiap kata yang dimaksud, n merupakan jumlah keseluruhan dokumen, dan n_k adalah jumlah kemunculan *term* pada keseluruhan dokumen. Setelah mendapatkan nilai TF dan IDF, maka tahap selanjutnya adalah mendapatkan nilai TF-IDF dengan mengalikan nilai $TF_{t,I}$ dengan IDF_I. Berikut adalah hasil perhitungan TF-IDF pada tiap tahap yang dapat dilihat pada Tabel 3.3.

Tabel 3.2. Hasil perhitungan TF-IDF

Term	TF-IDF		*Term*	TF-IDF	
	q	d1		q	d1
guna	0,0000	0,0000	grafis	0,0000	0,0000
kontrol	0,0000	0,0000	rupa	0,0000	0,0000
tekan	0,0000	0,0000	fungsi	-0,0048	-0,0077
tampil	-0,0095	0,0000	input	-0,0048	-0,0077
bahasa	0,0000	0,0000	gera	0,0000	0,0000
pilih	0,0000	0,0000	indonesia	0,0000	0,0000
gembung	0,0000	0,0000	click	0,0000	0,0000
scroll	0,0000	0,0000	file	0,0000	0,0000
tetikus	0,0000	0,0000	interface	0,0000	0,0000
ikon	0,0000	0,0000	kendali	0,0000	0,0000
user	0,0000	0,0000	menaikturunkan	0,0000	0,0000
gerak	0,0000	0,0000	mouse	-0,0190	-0,0077
arti	0,0000	0,0000	tombol	0,0000	0,0000
folder	0,0000	0,0000	kabel	0,0000	0,0000
teks	0,0000	0,0000	komputer	-0,0048	-0,0077
perangkat	-0,0048	-0,0077	layar	-0,0048	-0,0153
monitor	0,0000	0,0000	keras	0,0000	0,0000
kursor	-0,0095	-0,0077	pindah	0,0000	0,0000
bentuk	0,0000	0,0000	graphical	0,0000	0,0000
wireless	0,0000	0,0000	tikus	0,0000	0,0000
arah	0,0000	0,0000	badan	0,0000	0,0000

Term	**TF-IDF**		***Term***	**TF-IDF**	
	q	**d1**		**q**	**d1**
antarmuka	0,0000	0,0000	masuk	0,0000	0,0000
gui	-0,0048	-0,0077	naikturun	0,0000	0,0000

3.2.3.2 Menghitung Kesamaan Dengan *Cosine Similarity*

Setelah mendapatkan bobot pada kata, tahap selanjutnya adalah mencari nilai similaritas dokumen dengan menggunakan *cosine similarity*. *Query* yang digunakan adalah kunci jawaban yang telah melalui tahap pre-processing dan pembobotan TF-IDF. Perhitungan skalar dilakukan jika ditemukan *term* pada inputan jawaban dengan mengalikan W_{qi} dengan W_{dij}.

Tabel 3.3. Perhitungan *Cosine Similarity*

Term	**TF*IDF**		**w (qk x dk)**	**Kuadrat**	
	q	**d1**	**q.d1**	**q^2**	**$d1^2$**
guna	0,00000	0,00000	0,00000	0,00000	0,00000
kontrol	0,00000	0,00000	0,00000	0,00000	0,00000
tekan	0,00000	0,00000	0,00000	0,00000	0,00000
tampil	-0,00952	0,00000	0,00000	0,00009	0,00000
bahasa	0,00000	0,00000	0,00000	0,00000	0,00000
pilih	0,00000	0,00000	0,00000	0,00000	0,00000
gembung	0,00000	0,00000	0,00000	0,00000	0,00000
scroll	0,00000	0,00000	0,00000	0,00000	0,00000
tetikus	0,00000	0,00000	0,00000	0,00000	0,00000
ikon	0,00000	0,00000	0,00000	0,00000	0,00000
user	0,00000	0,00000	0,00000	0,00000	0,00000
gerak	0,00000	0,00000	0,00000	0,00000	0,00000
arti	0,00000	0,00000	0,00000	0,00000	0,00000
folder	0,00000	0,00000	0,00000	0,00000	0,00000
teks	0,00000	0,00000	0,00000	0,00000	0,00000
perangkat	-0,00476	-0,00766	0,00004	0,00002	0,00006
monitor	0,00000	0,00000	0,00000	0,00000	0,00000
kursor	-0,00952	-0,00766	0,00007	0,00009	0,00006
bentuk	0,00000	0,00000	0,00000	0,00000	0,00000
wireless	0,00000	0,00000	0,00000	0,00000	0,00000
arah	0,00000	0,00000	0,00000	0,00000	0,00000
antarmuka	0,00000	0,00000	0,00000	0,00000	0,00000
grafis	0,00000	0,00000	0,00000	0,00000	0,00000
rupa	0,00000	0,00000	0,00000	0,00000	0,00000
fungsi	-0,00476	-0,00766	0,00004	0,00002	0,00006
input	-0,00476	-0,00766	0,00004	0,00002	0,00006

Term	TF*IDF		w (qk x dk)	Kuadrat	
	q	d1	q.d1	q^2	$d1^2$
gera	0,00000	0,00000	0,00000	0,00000	0,00000
indonesia	0,00000	0,00000	0,00000	0,00000	0,00000
click	0,00000	0,00000	0,00000	0,00000	0,00000
file	0,00000	0,00000	0,00000	0,00000	0,00000
interface	0,00000	0,00000	0,00000	0,00000	0,00000
kendali	0,00000	0,00000	0,00000	0,00000	0,00000
menaikturunkan	0,00000	0,00000	0,00000	0,00000	0,00000
mouse	-0,01904	-0,00766	0,00015	0,00036	0,00006
tombol	0,00000	0,00000	0,00000	0,00000	0,00000
kabel	0,00000	0,00000	0,00000	0,00000	0,00000
komputer	-0,00476	-0,00766	0,00004	0,00002	0,00006
layar	-0,00476	-0,01531	0,00007	0,00002	0,00023
keras	0,00000	0,00000	0,00000	0,00000	0,00000
pindah	0,00000	0,00000	0,00000	0,00000	0,00000
graphical	0,00000	0,00000	0,00000	0,00000	0,00000
tikus	0,00000	0,00000	0,00000	0,00000	0,00000
badan	0,00000	0,00000	0,00000	0,00000	0,00000
masuk	0,00000	0,00000	0,00000	0,00000	0,00000
naikturun	0,00000	0,00000	0,00000	0,00000	0,00000
gui	-0,00476	-0,00766	0,00004	0,00002	0,00006
		Jumlah	**0,000474**	**0,000680**	**0,000645**

Selanjutnya, nilai panjang pada setiap dokumen termasuk *query* dengan mengkuadratkan bobot dari *query* dengan bobot dari dokumen jawaban lalu dijumlahkan dan hasil penjumlahan akan diakarkan. Hasil perkalian skalar akan dibagi dengan hasil panjang vektor untuk mendapatkan hasil similaritas antara jawaban dengan kunci jawaban. Berdasarkan Rumus 2.4, perhitungan *Cosine Similarity* dapat dilakukan seperti sebagai berikut.

$$Cosine\ Similarity(q, d1) = \frac{\sum_{k=1}^{t} w_{qk} \times w_{dk}}{\sqrt{\sum_{k=1}^{t}(w_{qk})^2} \cdot \sqrt{\sum_{k=1}^{t}(w_{dk})^2}}$$

$$Cosine\ Similarity(q, d1) = \frac{0{,}000474}{\sqrt{0{,}000680}\sqrt{0{,}000645}}$$

$$Cosine\ Similarity(q, d1) = 0{,}7156264473$$

3.2.3.3 Konversi Skor

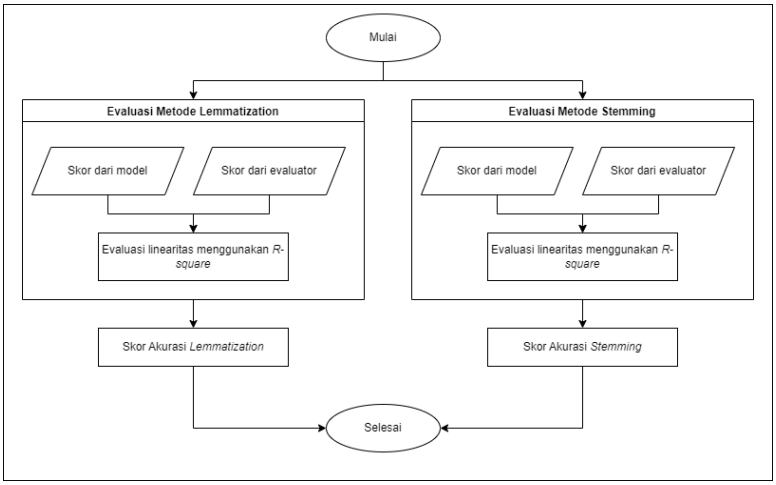
Hasil dari perhitungan *cosine similarity* berentang dari 0 hingga 1. *Scoring* dilakukan untuk menghitung bobot hasil nilai *cosine similarity* dikali dengan bobot soal. Bobot soal didapatkan dari ketentuan yang diberikan oleh evaluator. Misalkan setiap soal memiliki bobot maksimal 100, maka hasil yang telah didapatkan dari tahap sebelumnya yaitu *Cosine Similarity* dikalikan dengan bobot soal. Nilai skor yang didapatkan nantinya adalah total dari skor tiap soal tersebut dibagi dengan jumlah soal untuk masing-masing siswa. Untuk simulasi perhitungan konversi skor dengan jumlah soal 5 pada model ditunjukkan pada Tabel 3.5 berikut.

Tabel 3.4. Konversi Nilai

Nomor	**Bobot Soal**	**D1**	**D2**	**Skor per-soal**	
				D1	**D2**
1	100	0,234	0,67	23,4	67
2	100	0,52	0,532	52	53,2
3	100	0,9	0,123	90	12,3
4	100	0,5	0,743	50	74,3
5	100	0,444	0,634	44,4	63,4
			Total	259,8	270,2
			Skor	**51,96**	**54,04**

3.2.4 Evaluasi

Tahap evaluasi merupakan tahap akhir dalam penelitian untuk mengevaluasi perbandingan antara hasil dari model Penilaian Jawaban Pendek Otomatis dengan jawaban evaluator pada kedua metode *stemming* dan metode *lemmatization*. Adapun alur dari evaluasi kedua mode model Penilaian Esai Pendek Otomatis menggunakan penerapan metode *lemmatization* dan *stemming* dapat dilihat pada Gambar 3.20 berikut.



Gambar 3.20 Alur tahap evaluasi model *stemming* dan *lemmatization*

Setelah sistem selesai memberi skor semua jawaban siswa, maka tahap selanjutnya adalah mengukur akurasi hasil sistem dengan penilaian manual dari evaluator menggunakan korelitas determinan. Contoh perhitungan dari 10 data skor sistem dan skor evalutor ditunjukkan pada tabel 3.6.

Tabel 3.5. Perhitungan evaluasi *Mean Absolute Error*

No.	**Skor Sistem (X)**	**Skor Evaluator (Y)**	**|X-Y|**
1	80	85	5
2	70	65	5
3	40	70	30
4	10	35	25
5	100	100	0
6	20	25	5
7	70	75	5
8	70	77	7
9	50	65	15
10	80	41	39
		Jumlah	**136**

Setelah mendapatkan jumlah dari nilai mutlak dari *error*, maka nilai dari *Mean Absolute Errot* sudah dapat dihitung sesuai dengan Rumus 2.2 seperti sebagai berikut.

$$MAE = \frac{\sum_{i=1}^{n}|(X_i - Y_i)|}{n}$$

$$MAE = \frac{136}{10} = 13{,}6$$

Hasil perhitungan menggunakan MAE memiliki rentang dari 0 hingga ∞. Sehinnga, semakin dekat hasil dengan 0, maka semakin baik performa dari model. Dari perhitungan diatas, didapat nilai MAE sebesar 13,6, maka dapat diketahui terdapat perbedaan sebesar 13,6.

3.3 Alat dan Bahan Tugas Akhir

3.3.1 Alat

1. *Notebook* dengan spesifikasi sistem operasi Windows 11, *processor* AMD Ryzen 5 3550H CPU @ 2,1 GHz, memori 8GB DDR4, grafis AMD Radeon RX 560X (4GB), *hardisk* 1TB.
2. Jupyter *Notebook* dengan Python versi 3.8 dengan modul sklearn, numpy, pandas, nltk, Sastrawi.
3. *Source code* kamus morfologikal bahasa Melayu/Indonesia MALINDO Morph dari https://github.com/matbahasa/MALINDO_Morph

3.3.2 Bahan

Bahan yang digunakan untuk melakukan penelitian adalah sebagai berikut:

1. Dataset berlisensi CC-BY 4.0 yang dibuat oleh Rahutomo dengan nama ”*Indonesian Query Answering Dataset for Online Essay Test System*” pada tahun 201. Dataset berisikan 4 kategori yaitu politik, gaya hidup, olahraga, dan teknologi, namun pada penelitian ini kategori yang digunakan hanya kategori teknologi. Masing-masing kategori memiliki 10 pertanyaan dengan bobot skor tiap soal adalah 1-100 [18]. Soal dan kunci jawaban yang digunakan pada dataset dapat dilihat pada Tabel 3.8.

Tabel 3.8 Dataset soal dan kumci jawaban

No	Soal	Kunci Jawaban
1	Apa yang dimaksud dengan komputer?	Komputer adalah rangkaian mesin elektronik yang dapat bekerja sama. Sistem ini digunakan untuk memudahkan pekerjaan

No	Soal	Kunci Jawaban
	(Jawab dalam 1-3 kalimat)	manusia. Komputer bekerja otomatis berdasarkan urutan instruksi atau program yang diberikan.
2	Sebutkan keuntungan-keuntungan yang bisa diperoleh dari penerapan teknologi dalam kegiatan bisnis? (Sebutkan minimal 3)	- Memperluas pemasaran sehingga tidak mempermasalahkan jarak dan waktu - Mengurangi biaya produksi, promosi - Mempermudah penyimpanan data penjualan, data barang maupun laporan keuangan - Menggantikan pekerjaan manual menjadi otomatis - Proses produksi lebih cepat dan praktis
3	Jelaskan yang dimaksud dengan hardware dan berikan contohnya (minimal 5).	Hardware (perangkat keras) adalah suatu komponen yang ada pada komputer, bisa dilihat secara kasat mata dan dapat disentuh secara fisik. Contoh : mouse, monitor, keyboard, printer, kamera, cpu, kabel, router, bridge, hub.
4	Apa perbedaan dari data, informasi dan pengetahuan?	Data merupakan rangkaian fakta yang mewakili suatu kejadian, atau dapat di sebut dengan simbol yang terekam. Informasi adalah makna yang dapat di tafsirkan dari data. Pengetahuan adalah informasi yang telah dimaknai sehingga dapat digunakan untuk meningkatkan pengertian atau pemahaman.
5	Bagaimana cara mengatasi sampah elektronik (e-waste)? (Sebutkan minimal 3)	- Mendaur ulang sampah elektronik menjadi barang yang berguna dan memiliki nilai jual - Memisahkan barang elektronik sesuai komposisi bahannya - tidak membuang sembarangan sehingga karena dapat mencemari lingkungan - Dikembalikan kepada produsen elektronik
6	Apa yang dimaksud dengan volatile memory?	Volatile memory adalah memory yang datanya dapat ditulis dan dihapus, tetapi hilang saat kehilangan power (kondisi off atau mati lampu).
7	Apa kepanjangan dari LCD, CPU dan GPS?	LCD (Liquid Crystal Display), CPU (Central Processing Unit), GPS (Global Positioning System)
8	Sebutkan nama versi android dari Cupcake hingga saat ini.	Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat dan Lolypop.
9	Apa yang dimaksud topologi jaringan LAN dan sebutkan contoh topologi tersebut (minimal 3).	Topologi jaringan adalah struktur jaringan fisik yang digunakan untuk implementasi LAN (local area network). Unsur dasar penyusun jaringan, yaitu node, link, dan station. Contoh Topologi bintang (star) ,

No	Soal	Kunci Jawaban
		Topologi cincin (ring) , Topologi bus, Topologi pohon (tree), Topologi linier
10	Apa yang dimaksud dengan Bluetooth?	Bluetooth adalah perangkat yang menjadi media tukar menukar (menerima mengirim) informasi di antara peralatan eketronik. Bluetooth merupakan media tanpa kabel. Biasanya bluethooth digunakan untuk mengirim foto atau file antar handphone.

Setiap pertanyaan memiliki 50-52 jawaban yang berbeda serta nilai manual dari evaluator. Pada dataset terdapat tiga nilai manual berbeda, sehingga nilai manual yang digunakan pada penelitian ini adalah rata-rata dari ketiga nilai manual tersebut. Contoh data jawaban siswa yang digunakan ditunjukkan pada Tabel 3.9

Tabel 3.9 Dataset soal dan kunci jawaban

No	Siswa	Jawaban	Manual
1	siswa_1	Komputer adalah serangkaian ataupun sekelompok mesin elektronik yang terdiri dari ribuan bahkan jutaan komponen yang dapat saling bekerja sama, serta membentuk sebuah sistem kerja yang rapi dan teliti. Sistem ini kemudian dapat digunakan untuk melaksanakan serangkaian pekerjaan secara otomatis, berdasar urutan instruksi ataupun program yang diberikan kepadanya.	76,66667
2	siswa_2	komputer adalah mesin penghitung	11,66667
3	siswa_3	mesin yang membantu manusia untuk menjalankan segara pekerjaan	25

2. Dokumen acuan yang penelitian terkait yang telah didapat dan disitasi peneliti untuk mendukung penelitian.

DAFTAR PUSTAKA

[1] "Badan Pusat Statistik." https://www.bps.go.id/indikator/indikator/view_data_pub/0000/api_pub/dnF4TTdwbEcwbTFHazAwZUtOMVRBQT09/da_04/1 (accessed Jul. 12, 2022).

[2] "Badan Pusat Statistik." https://www.bps.go.id/indikator/indikator/view_data_pub/0000/api_pub/a1lFcnlHNXNYMFlueG8xL0ZOZnU0Zz09/da_04/1 (accessed Jul. 12, 2022).

[3] "Badan Pusat Statistik." https://www.bps.go.id/indikator/indikator/view_data_pub/0000/api_pub/UkJNaEl6ZHRVYXNaMzZhZG9BbS9ZZz09/da_04/1 (accessed Jul. 12, 2022).

[4] "Badan Pusat Statistik." https://www.bps.go.id/indikator/indikator/pencarian?keyword=SMP (accessed Jul. 12, 2022).

[5] R. N. A. Sanusi and F. Aziez, "Analisis Butir Soal Tes Objektif dan Subjektif untuk Keterampilan Membaca Pemahaman pada Kelas VII SMP N 3 Kalibagor," *Metaf. J. Pembelajaran Bhs. Dan Sastra*, vol. 8, no. 1, p. 99, 2021, doi: 10.30595/mtf.v8i1.8501.

[6] V. R. Prasetyo, M. Widiasri, and M. M. Angkiriwang, "Sistem Berbasis Web Untuk Koreksi Soal Esai Dengan Association Rules," *Teknika*, vol. 11, no. 1, pp. 62–68, 2022, doi: 10.34148/teknika.v11i1.449.

[7] M. W. Arbiantono, "Pengembangan Aplikasi Assessment Tool Menggunakan Metode Cosine Semantic Similarity untuk Automatic Scoring Jawaban Tes Uraian pada Mata Pelajaran Basis Data di SMKN 1 Surabaya," *Inf. Technol. Educ.*, vol. 5, no. 2, pp. 657–666, 2021.

[8] M. Nurjannah and I. Fitri Astuti, "PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING Mahasiswa S1 Program Studi Ilmu Komputer FMIPA Universitas Mulawarman Dosen Program Studi Ilmu Komputer FMIPA Universitas Mulawarman," *J. Inform. Mulawarman*, vol. 8, no. 3, pp. 110–113, 2013.

[9] N. Vendyansyah and Y. A. Pranoto, "Perancangan dan Pembuatan Aplikasi Untuk Mendeteksi Kemiripan Jawaban Menggunakan Metode Cosine

Similarity."

[10] M. Kunilovskaya and A. Plum, "Text Preprocessing and its Implications in a Digital Humanities Project," *Int. Conf. Recent Adv. Nat. Lang. Process. RANLP*, vol. 2021-Septe, pp. 85–93, 2021, doi: 10.26615/issn.2603-2821.2021_013.

[11] V. Balakrishnan and L.-Y. Ethel, "Stemming and Lemmatization: A Comparison of Retrieval Performances," *Lect. Notes Softw. Eng.*, vol. 2, no. 3, pp. 262–267, 2014, doi: 10.7763/lnse.2014.v2.134.

[12] K. Sirts and K. Peekman, "Evaluating sentence segmentation and word tokenization systems on estonian web texts," *Front. Artif. Intell. Appl.*, vol. 328, pp. 174–181, 2020, doi: 10.3233/faia200620.

[13] "IMPLEMENTASI METODE NAIVE BAYES CLASSIFIER UNTUK APLIKASI FILTERING EMAIL SPAM DENGAN LEMMATIZATION BERBASIS WEB | , Rahmi Hidayati | Coding Jurnal Komputer dan Aplikasi." https://jurnal.untan.ac.id/index.php/jcskommipa/article/view/25487/75676576644 (accessed Jul. 13, 2022).

[14] N. Chamidah and M. M. Santoni, "Pencocokan Berbasis Kata Kunci pada Penilaian Esai Pendek Otomatis Berbahasa Indonesia," *Techno.Com*, vol. 20, no. 1, pp. 19–27, 2021, doi: 10.33633/tc.v20i1.4115.

[15] K. Tuomo *et al.*, "Authors : Stemming and lemmatization in the clustering of finnish text conference on Information and knowledge management Editors of work : Pages : Stemming and Lemmatization in the Clustering of Finnish Text Documents," 2004.

[16] I. Boban, A. Doko, and S. Gotovac, "Sentence retrieval using Stemming and Lemmatization with different length of the queries," *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 3, pp. 349–354, 2020, doi: 10.25046/aj050345.

[17] F. M. Ichsan, "PENGARUH PENGGUNAAN STEMMING DAN LEMMATIZATION TERHADAP AKURASI ANALISIS SENTIMEN," Jul. 2020.

[18] F. RAHUTOMO and T. ARI ROSHINTA, "Indonesian Query Answering Dataset for Online Essay Test System," vol. 1, 2018, doi: 10.17632/6GP8M72S9P.1.

[19] U. Hasanah, T. Astuti, R. Wahyudi, Z. Rifai, and R. A. Pambudi, "An

experimental study of text preprocessing techniques for automatic short answer grading in Indonesian," *Proc. - 2018 3rd Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. ICITISEE 2018*, pp. 230–234, 2018, doi: 10.1109/ICITISEE.2018.8720957.

[20] A. R. Lahitani, "Automated Essay Scoring menggunakan Cosine Similarity pada Penilaian Esai Multi Soal," *J. Kaji. Ilm.*, vol. 22, no. 2, pp. 107–118, 2022, doi: 10.31599/jki.v22i2.1121.

[21] M. Alobed, A. M. M. Altrad, and Z. B. A. Bakar, "A Comparative Analysis of Euclidean, Jaccard and Cosine Similarity Measure and Arabic Wordnet for Automated Arabic Essay Scoring," *Proc. - CAMP 2021 2021 5th Int. Conf. Inf. Retr. Knowl. Manag. Digit. Technol. IR 4.0 Beyond*, pp. 70–74, 2021, doi: 10.1109/CAMP51653.2021.9498119.

[22] M. Shermis and B. JC, "Automated Essay Scoring: A Cross-Disciplinary Perspective," *Mahwah. New Jersey London Lawrence Erlbaum Assoc.*, 2003.

[23] H. V. Nguyen and D. J. Litman, "Argument mining for improving the automated scoring of persuasive essays," *32nd AAAI Conf. Artif. Intell. AAAI 2018*, no. md, pp. 5892–5899, 2018, doi: 10.1609/aaai.v32i1.12046.

[24] S. Burrows, I. Gurevych, and B. Stein, *The eras and trends of automatic short answer grading*, vol. 25, no. 1. 2015.

[25] C. M. Ormerod, A. Malhotra, and A. Jafari, "Automated essay scoring using efficient transformer-based language models," pp. 1–11, 2021, [Online]. Available: http://arxiv.org/abs/2102.13136.

[26] F. Rahimi and A. N. Asyikin, "Aplikasi Penilaian Ujian Essay Otomatis Menggunakan Metode Cosine Similarity," *Poros Tek.*, vol. 7, no. 2, pp. 88–94, 2015, [Online]. Available: http://ejurnal.poliban.ac.id/index.php/porosteknik/article/view/218.

[27] "What is Natural Language Processing (NLP)?" https://www.textmetrics.com/what-is-natural-language-processing-nlp (accessed Jul. 14, 2022).

[28] Yuliana, *Pengolahan Bahasa Alami*. 2014.

[29] V. I. Santoso[1], G. Virginia[2], and Y. Lukito[3], "Penerapan Sentimen Analisis Pada Hasil Evaluasi Dosen Dengan Metode SVM," *J. Transform.*, vol. 14, no. 1, pp.

72–76, 2017.

[30] Schütze, Hinrich, and P. Manning, Christopher D. Raghavan, *Introduction to information retrieval*, vol. 39. Cambridge, 2008.

[31] R. Riyaddulloh and A. Romadhony, "Normalisasi Teks Bahasa Indonesia Berbasis Kamus Slang Studi Kasus: Tweet Produk Gadget Pada Twitter," *eProceedings Eng.*, vol. 8, no. 4, pp. 4216–4228, 2021, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15246/14969.

[32] "eLearning/slang.dic at master · taudataid/eLearning." https://github.com/taudataid/eLearning/blob/master/data/slang.dic (accessed Dec. 13, 2022).

[33] D. Suhartono, D. Christiandy, and R. Rolando, "Lemmatization Technique in Bahasa: Indonesian Language," *J. Softw.*, vol. 9, no. 5, 2014, doi: 10.4304/jsw.9.5.1202-1209.

[34] H. Nomoto, H. Choi, D. Moeljadi, and F. Bond, "MALINDO Morph: Morphological dictionary and analyser for Malay/Indonesian," *Proc. Elev. Int. Conf. Lang. Resour. Eval. (LREC 2018)*, pp. 36–43, 2018.

[35] H. R. Pramudita, "Penerapan Algoritma Stemming Nazief & Adriani Dan Similarity Pada Penerimaan Judul Thesis," *J. Ilm. Data Manaj. dan Teknol. Inf.*, vol. 15, no. 4, pp. 15–19, 2014.

[36] S. B. S, D. Khyani, N. N. M, and D. B. M, "An Interpretation of Lemmatization and Stemming in Natural Language Processing," *J. Univ. Shanghai Sci. Technol.*, vol. 22, no. 10, pp. 350–357, 2020, [Online]. Available: https://www.researchgate.net/publication/348306833.

[37] L. Agusta, "Stemming Porter Dengan Untuk Stemming Dokumen Teks Bahasa Indo ...," *Proc. Konf. Nas. Sist. dan Inform.*, pp. 196–201, 2009.

[38] "sastrawi/sastrawi: [Inactive] High quality stemmer library for Indonesian Language (Bahasa)." https://github.com/sastrawi/sastrawi (accessed Oct. 24, 2022).

[39] J. Asian, H. E. Williams, and S. M. M. Tahaghoghi, "Stemming Indonesian," *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 38, no. January, pp. 307–314, 2005, doi: 10.1145/1316457.1316459.

[40] A. Jelita, “Effective Techniques for Indonesian Text Retrieval,” *Ph.D Thesis*, pp. 1–286, 2007, [Online]. Available: https://researchbank.rmit.edu.au/view/rmit:6312.

[41] A. Z. Arifin, P. A. D. . Mahendra, and H. T. Ciptaningtyas, “Enhanced Confix Stripping Stemmer And Ants Algorithm for Classifying News Document in Indonesian Language,” *5thInternational Conf. Inf. Commun. Technol. Syst.*, no. April 2014, pp. 149–158, 2009.

[42] D. P. Andita Dwiyoga Tahitoe, “Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming,” *J. Ilm.*, pp. 1–15, 2010.

[43] A. Deolika, K. Kusrini, and E. T. Luthfi, “Analisis Pembobotan Kata Pada Klasifikasi Text Mining,” *J. Teknol. Inf.*, vol. 3, no. 2, p. 179, 2019, doi: 10.36294/jurti.v3i2.1077.

[44] E. Prayitno, T. Suprawoto, and B. F. Riyanto, “OPTIMASI HASIL PENCARIAN PADA WEB SCRAPPING MENGGUNAKAN PEMBOBOTAN KATA TF-IDF,” vol. 1, no. 7, 2021.

[45] P. M. Hasugian, J. Manurung, L. Logaraz, and U. Ram, “Implementation of Tf-Idf and Cosine Similarity Algorithms for Classification of Documents Based on Abstract Scientific Journals,” *Infokum*, vol. 9, no. 2, pp. 518–526, 2021, [Online]. Available: http://infor.seaninstitute.org/index.php/infokum/article/view/201%0Ahttp://infor.seaninstitute.org/index.php/infokum/article/download/201/144.

[46] R. R. Et.al, “The Similarity of Essay Examination Results using Preprocessing Text Mining with Cosine Similarity and Nazief-Adriani Algorithms,” *Turkish J. Comput. Math. Educ.*, vol. 12, no. 3, pp. 1415–1422, 2021, doi: 10.17762/turcomat.v12i3.938.

[47] C. J. Willmott and K. Matsuura, “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance,” *Clim. Res.*, vol. 30, no. 1, pp. 79–82, 2005, doi: 10.3354/cr030079.

[48] M. Razzaghi, G. J. McLachan, and K. E. Basford, “Mixture Models,” *Technometrics*, vol. 33, no. 3, p. 365, 1991, doi: 10.2307/1268796.