

**PERBANDINGAN LEMMATIZATION DAN STEMMING
PADA PENILAIAN ESAI SINGKAT BERBAHASA
INGGRIS MENGGUNAKAN TF-IDF DAN COSINE
SIMILARITY**

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:

APRILIA PURWANTO

119140003



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI, PRODUKSI DAN INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN**

2022

LEMBAR PENGESAHAN

Tugas Akhir dengan judul “Perbandingan Lemmatization dan Stemming Pada Penilaian Esai Singkat Berbahasa Inggris 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

Aprilia Purwanto
NIM. 119140003

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

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

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir dengan judul “Perbandingan Lemmatization dan Stemming Pada Penilaian Esai Singkat Berbahasa Inggris Menggunakan TF-IDF dan Cosine Similarity” adalah karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan benar.

Nama : Aprilia Purwanto

NIM : 119140003

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 : Aprilia Purwanto
NIM : 119140003
Program Studi : Teknik Informatika
Jurusan : Jurusan Teknologi, Produksi dan Industri
Jenis Karya : Tugas Akhir

demikian 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 Lemmatization dan Stemming Pada Penilaian Esai Singkat Berbahasa Inggris Menggunakan TF-IDF dan Cosine Similarity”

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

Yang menyatakan,

Aprilia Purwanto

KATA PENGANTAR

Puji syukur kehadiran 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 mengucapkan 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 dan Kakak 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

Judul TA

Nama Mahasiswa

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

Judul TA

Nama Mahasiswa

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

Judul TA (Bahasa Inggris)

Nama Mahasiswa

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

LEMBAR PENGESAHAN	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	iv
KATA PENGANTAR	v
RINGKASAN	vi
ABSTRAK.....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR RUMUS	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah.....	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	6
2.1 Tinjauan Pustaka	6
2.2 Dasar Teori.....	11
2.2.1 Ujian Esai Singkat.....	11

2.2.2	Penilaian Esai Otomatis	12
2.2.3	Text Mining.....	12
2.2.4	Natural Language Processing (NLP)	13
2.2.5	Preprocessing	13
2.2.6	Case Folding.....	13
2.2.7	Tokenizing.....	13
2.2.8	POS Tagging	14
2.2.9	Stopword Removal.....	15
2.2.10	Stemming	15
2.2.11	Lemmatization.....	16
2.2.12	TF-IDF	17
2.2.13	Cosine Similarity.....	18
2.2.14	Mean Absoulte Error (MAE)	19
BAB III METODE PENELITIAN		20
3.1	Alur Penelitian	20
3.2	Penjabaran Langkah Penelitian	21
3.2.1	Studi Literatur	21
3.2.2	Pengumpulan Data	21
3.2.3	Perancangan Model	21
3.2.4	Analisis Hasil	33
3.3	Alat dan Bahan Tugas Akhir.....	35
3.3.1	Alat.....	35
3.3.2	Bahan.....	35
DAFTAR PUSTAKA		36

DAFTAR TABEL

Tabel 2.1 Perbandingan Referensi Penelitian Terdahulu.....	6
Tabel 2.2 <i>POS Tags Penn Treebank Project</i>	243
Tabel 3.1 Jawaban Peserta Didik	23
Tabel 3.2 Jawaban Ahli	23
Tabel 3.3 <i>Case Folding</i> Jawaban Peserta Didik	23
Tabel 3.4 <i>Case Folding</i> Jawaban Ahli	24
Tabel 3.5 <i>Tokenizing</i> Jawaban Peserta Didik	31
Tabel 3.6 <i>Tokenizing</i> Jawaban Ahli	31
Tabel 3.7 <i>Stopword Removal</i> Jawaban Peserta Didik.....	315
Tabel 3.8 <i>Stopword Removal</i> Jawaban Ahli	315
Tabel 3.9 <i>Stemming</i> Jawaban Peserta Didik	315
Tabel 3.10 <i>Stemming</i> Jawaban Ahli.....	316
Tabel 3.11 <i>POS Tagging</i> Jawaban Peserta Didik	316
Tabel 3.12 <i>POS Tagging</i> Jawaban Ahli.....	316
Tabel 3.13 <i>Lemmatization</i> Jawaban Peserta Didik	317
Tabel 3.14 <i>Lemmatization</i> Jawaban Ahli.....	317
Tabel 3.15 <i>TF-IDF Lemmatization</i>	318
Tabel 3.16 <i>TF-IDF Stemming</i>	318
Tabel 3.17 <i>Cosine Similarity Lemmatization</i>	31
Tabel 3.18 <i>Cosine Similarity Stemming</i>	32
Tabel 3.19 Konversi Skor Jawaban Siswa dengan <i>Lemmatization</i>	33
Tabel 3.20 Konversi Skor Jawaban Siswa dengan <i>Stemming</i>	33

DAFTAR GAMBAR

Gambar 2.1 Proses Lematisasi.....	206
Gambar 2.2 Hierarki <i>WordNet</i>	207
Gambar 3.1 Alur Penelitian	20
Gambar 3.2 Alur Perancangan Model	22

DAFTAR RUMUS

Rumus 2. 1.....	188
Rumus 2. 2.....	188
Rumus 2. 3.....	199
Rumus 2. 4.....	199

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Era globalisasi mempengaruhi laju pendidikan dunia termasuk salah satunya Indonesia. Salah satu hal positif yang dibawa oleh globalisasi adalah ilmu pengetahuan dan teknologi dari negara lain baik berupa buku atau artikel yang dapat diakses melalui internet. Sebagian besar referensi pembelajaran yang tersedia ini biasanya menggunakan istilah dan bahasa Inggris sebagai bahasa internasional [1]. Tujuannya adalah agar masyarakat dunia mampu memahami perkembangan ilmu pengetahuan dan teknologi di dunia. Oleh karena itu, agar perkembangan pendidikan dan teknologi Indonesia tidak tertinggal oleh dunia diperlukan pembelajaran dan penguasaan bahasa Inggris. Sama seperti proses pembelajaran pada bidang ilmu lain, diperlukan metode yang tepat agar pembelajaran bahasa Inggris dapat dilaksanakan dengan maksimal.

Untuk menunjang proses pembelajaran tersebut, diperlukan metode evaluasi untuk mengukur kemampuan peserta didik. Evaluasi biasanya diberikan oleh pengajar melalui pemberian tugas atau ujian berbentuk soal objektif dan soal esai. Soal objektif lebih mudah diterapkan karena berbentuk pilihan ganda, pencocokan, atau benar salah sehingga proses penilaiannya mudah dan cepat. Namun, kekurangannya adalah sulitnya mengetahui tingkat kemampuan peserta didik saat menulis dalam bahasa Inggris [2]. Untuk mengatasinya, soal berbentuk esai dirasa lebih tepat untuk diterapkan. Sarana ini memberikan kebebasan pada peserta didik untuk menuliskan ide, opini, dan solusi terhadap sebuah permasalahan dengan kreatif dalam sebuah uraian terbuka [3]. Peserta didik ditantang untuk dapat mengekspresikan opini mereka dengan pemilihan kata dan tata bahasa yang tepat sesuai dengan kaidah bahasa Inggris.

Meskipun demikian, jika dibandingkan dengan bentuk soal objektif maka proses penilaian bentuk soal esai akan lebih sulit dilakukan. Berbeda dengan soal tipe objektif, soal esai tidak memiliki pilihan jawaban. Peserta didik menjawab soal ini berdasarkan pengetahuannya sehingga akan muncul banyak variasi jawaban [4]. Pada akhirnya, evaluasi ini akan memerlukan waktu dan tenaga lebih. Selain itu, konsistensi penilai dapat menjadi masalah lain dalam evaluasi jawaban soal esai. Inkonsistensi penilaian dapat terjadi jika penilai melakukan evaluasi di waktu yang berbeda. Penilai

yang berbeda juga dapat menyebabkan hasil evaluasi yang berbeda-beda karena cara pandangan seorang penilai dalam melakukan evaluasi jawaban pasti berbeda-beda antar satu penilai dengan penilai lainnya [5]. Oleh karena itu, dibutuhkan suatu evaluator otomatis untuk mengevaluasi soal esai agar hasil penilaian tetap konsisten dan cepat.

Pengembangan sistem penilaian esai pendek berbahasa Inggris secara otomatis yang telah dilakukan oleh beberapa penelitian terbagi menjadi beberapa kelompok utama yaitu klasterisasi, similaritas dokumen, pembelajaran mesin, ekstraksi informasi dan pencocokan pola, serta pengolahan bahasa alami atau Natural Language Processing (NLP) [5]. Pada penelitian terdahulu, pengembangan sistem ini meliputi penggunaan beberapa metode seperti pembobotan TF-IDF, *term frequency*, *n-gram*, *cosine similarity*, dan *Pearson Correlation*. Setelahnya, akan dilakukan perbandingan antara dokumen jawaban milik siswa dan dokumen jawaban ahli. Selain cara tersebut, dapat juga dilakukan ekstraksi kata kunci dari kedua dokumen dan dinilai berdasarkan kecocokannya. Salah satu perpaduan metode yang umum digunakan untuk melakukan penilaian esai otomatis yaitu dengan melakukan pembobotan kata menggunakan TF-IDF (*Term Frequency-Inverse Document Frequency*) dan menghitung tingkat kemiripan antara jawaban uji dan jawaban ahli menggunakan *cosine similarity*.

Term Frequency Inverse Document Frequency (TF-IDF) merupakan metode yang digunakan untuk menghitung bobot suatu kata (*term*) pada suatu dokumen. Metode ini akan menghitung frekuensi kemunculan kata pada sebuah dokumen [6]. TF-IDF sudah dikenal sebagai metode yang mudah diterapkan, efisien, dan memiliki hasil yang akurat [7]. Lalu, digunakan perhitungan *cosine similarity* yang berfungsi untuk menghitung tingkat kesamaan antara dokumen uji dan dokumen ahli. Metode ini akan menghitung kata kunci (*keywords*) dari kedua dokumen dan menyatakannya dalam bentuk vektor [8]

Sebelum melakukan tahap pembobotan dan menghitung kemiripan, akan dilakukan tahap *text pre-processing* yang berguna untuk mengolah teks agar bersih dan siap untuk dianalisis lebih lanjut [9]. Dalam prosesnya, terdapat beberapa faktor yang perlu diperhatikan karena dapat menurunkan tingkat akurasi hasil pembobotan. Salah satunya yaitu dua kata bermakna sama bisa saja dideteksi berbeda karena memiliki imbuhan atau akhiran seperti kata “*work*” dan “*worked*” yang sama-sama

memiliki arti “kerja” namun hanya berbeda bentuk *tense*. Hal tersebut dapat diatasi dengan penggunaan *stemming* atau *lemmatization* yang berguna untuk menghasilkan kata ke bentuk awal (*root*) dengan menghilangkan *prefix* dan *suffix* dari sebuah kata. Perbedaan dari keduanya yakni *lemmatization* memastikan kata dasar yang dihasilkan merupakan kata yang memiliki makna [10].

Pada penelitian terdahulu, tahap *text pre-processing* banyak menggunakan proses *stemming* yang sebenarnya memiliki fungsi serupa dengan *lemmatization*. Namun, *lemmatization* sering dianggap lebih unggul karena dapat memastikan kata yang dihasilkan memiliki makna di dalam kamus [10]. Sebagai contoh, hasil dari *stemming* akan mengubah kata “*leaves*” menjadi “*leav*” sedangkan *lemmatization* akan mengubah “*leaves*” menjadi “*leaf*”. Belum banyak penelitian yang membandingkan hasil penggunaan kedua metode ini. Oleh sebab itu, perbandingan tingkat akurasi penggunaan *stemming* dan *lemmatization* khususnya pada penilaian esai otomatis menarik untuk diteliti lebih lanjut.

Berdasarkan permasalahan yang ada, penulis tertarik untuk melakukan penelitian terhadap perbandingan *lemmatization* dan *stemming* pada penilaian esai singkat berbahasa Inggris menggunakan metode *Term Frequency Inverse Document Frequency* (TF-IDF) dan *cosine similarity*.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah dijabarkan sebelumnya, dapat dirumuskan poin permasalahan yaitu :

1. Bagaimana tingkat akurasi penilaian esai singkat berbahasa Inggris menggunakan metode TF-IDF dan *cosine similarity* dengan penerapan *lemmatization*?
2. Bagaimana tingkat akurasi penilaian esai singkat berbahasa Inggris menggunakan metode TF-IDF dan *cosine similarity* dengan penerapan *stemming*?
3. Bagaimana perbandingan tingkat akurasi dengan menerapkan proses *stemming* dan proses *lemmatization* pada penilaian esai singkat berbahasa Inggris?

1.3 Tujuan Penelitian

Tujuan dari penelitian yang dilakukan yaitu :

1. Mengetahui tingkat akurasi penilaian esai singkat otomatis berbahasa Inggris menggunakan metode TF-IDF dan *cosine similarity* dengan penerapan *lemmatization*.
2. Mengetahui tingkat akurasi penilaian esai singkat otomatis berbahasa Inggris menggunakan metode TF-IDF dan *cosine similarity* dengan penerapan *stemming*.
3. Mengetahui perbandingan tingkat akurasi dengan menerapkan proses *stemming* dan proses *lemmatization* pada penilaian esai singkat berbahasa Inggris.

1.4 Batasan Masalah

Batasan penelitian yang dilakukan yaitu pemodelan penilaian esai pendek hanya pada teks berbahasa Inggris. Panjang ideal teks dokumen yang diproses model sama seperti panjang jawaban esai pendek yaitu sekitar 5-100 kata. Model hanya dapat memproses *input* data berupa huruf kapital (A-Z) dan huruf kecil (a-z) yang terdapat pada dokumen.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah dapat diterapkannya metode TF-IDF dan *cosine similarity* dengan membandingkan penerapan proses *stemming* dan *lemmatization* pada pemodelan penilaian esai pendek teks berbahasa Inggris. Selain itu, penelitian dapat dijadikan bahan referensi bagi penelitian selanjutnya dengan topik atau metode serupa yang terkait.

1.6 Sistematika Penulisan

1.6.1 Bab I – Pendahuluan

Bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan.

1.6.2 Bab II – Tinjauan Pustaka

Bab ini berisi tentang tinjauan pustaka dan dasar teori yang berkaitan dan dapat menunjang penelitian.

1.6.3 Bab III – Metode Penelitian

Bab ini berisi tentang alur penelitian, penjabaran langkah penelitian dari awal hingga akhir disertai ilustrasi perhitungan serta alat dan bahan penelitian.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pada penelitian ini, peneliti mendapatkan tinjauan pustaka dari penelitian-penelitian yang berkaitan dengan pembangunan model penilaian esai dan penggunaan metode TF-IDF, *Cosine similarity*, *stemming* dan *lemmatization*. Beberapa tinjauan tersebut dijadikan acuan dan pertimbangan dalam memilih metode yang digunakan dalam penelitian ini. Penelitian-penelitian yang berkaitan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan Referensi Penelitian Terdahulu

No	Penulis	Judul	Metode	Hasil	Persamaan
1	Alfina Rizqi Lahitani	“Automated Essay Scoring menggunakan <i>Cosine Similarity</i> pada Penilaian Esai Multi Soal”	<ol style="list-style-type: none"> 1. TF-IDF 2. <i>Cosine Similarity</i> 3. <i>Case folding</i> 4. <i>Stemming</i> 5. <i>Stopwords removal</i> 6. <i>Indexing</i> 	<p>Nilai akhir yang diperoleh sangat kecil untuk sebuah skor yang proporsional karena jika rata-rata pemberian nilai akhir berkisar antara 80-100, maka <i>cosine similarity</i> mewakili 52% poin dibandingkan dengan nilai aslinya dan tidak memiliki unsur subjektif.</p>	<p>Penggunaan TF-IDF. <i>cosine similarity</i>, tahap <i>text preprocessing case folding</i>, <i>stemming</i>, <i>stopword removal</i> pada objek <i>automated essay scoring</i>.</p>
2	Nurul Chamidah, dkk	“Penilaian Esai Pendek Otomatis dengan Pencocokan Kata Kunci Frasa Nomina”	<ol style="list-style-type: none"> 1. <i>Case folding</i> 2. <i>Part of Speech (POS) Tagging</i> 3. Ekstraksi frasa nomina 4. <i>Stemming</i> 	<p>Hasil evaluasi menunjukkan <i>Mean Absolute Error (MAE)</i> sebesar 18% dan <i>Pearson Correlation</i> sebesar 0.83.</p>	<p>Penggunaan tahap <i>text preprocessing</i> yaitu <i>case folding</i>, <i>stemming</i>, dan <i>POS tagging</i> pada objek penilaian esai</p>

No	Penulis	Judul	Metode	Hasil	Persamaan
			5. <i>Pearson correlation</i>		pendek otomatis.
3	Nurul Chamidah, dkk	“Pencocokan Berbasis Kata Kunci pada Penilaian Esai Pendek Otomatis Berbahasa Indonesia”	1. <i>Case folding</i> 2. <i>Stopword removal</i> 3. <i>Stemming</i> 4. <i>Tokenization</i> 5. <i>Keyword matching</i> 6. <i>Pearson correlation</i>	Hasil menunjukkan nilai <i>Mean Absolute Error</i> (MAE) sebesar 0.25 dan <i>Pearson Correlation</i> 0.79.	Penggunaan tahap <i>text preprocessing</i> yaitu <i>case folding</i> , <i>stopword removal</i> , <i>stemming</i> , dan <i>tokenization</i> pada objk penilaian esai pendek otomatis.
4	Ade Riyani, dkk	“Penerapan <i>Cosine Similarity</i> dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen”	1. TF-IDF 2. <i>Cosine similarity</i> 3. <i>Case folding</i> 4. <i>Stopword removal</i> 5. <i>Stemming</i> 6. <i>Tokenizing</i>	Hasil penelitian menunjukkan <i>cosine similarity</i> dan pembobotan TF-IDF telah berhasil mendeteksi kemiripan suatu dokumen dan proses <i>stemming</i> sangat berpengaruh pada nilai kemiripan hasil yakni menjadi lebih tinggi. Nilai rata-rata perbedaan nilai kemiripan jika menggunakan <i>stemming</i> dan tidak menggunakan <i>stemming</i> sebesar 10%.	Penggunaan metode TF-IDF, dan <i>Cosine similarity</i> serta tahap <i>text preprocessing</i> yaitu <i>case folding</i> , <i>stopword removal</i> , <i>tokenizing</i> , dan <i>stemming</i> .
5	Regi Ismayana Pratama, dkk	“ <i>Detector Similarity Answers Between Students on</i> ”	1. <i>Case folding</i> 2. <i>Tokenizing</i>	Murid dengan tingkat kemiripan 75% atau lebih akan dianggap	Penggunaan metode <i>Cosine similarity</i> dan tahap <i>text</i>

No	Penulis	Judul	Metode	Hasil	Persamaan
		<i>Essay Digital Exam System</i>	3. <i>Stopword remover</i> 4. <i>Stemming</i> 5. TF 6. N-Gram 7. <i>Cosine similarity</i>	mencontoh. Hasil penelitian menunjukkan 90% dari rekomendasi sistem sesuai dengan penilaian guru pada jawaban menyontek. Sistem mampu mendukung guru dalam mengevaluasi ujian atau belajar. N-gram dan <i>cosine similarity</i> merupakan metode yang baik untuk mendeteksi kemiripan jawaban murid.	<i>preprocessing</i> yaitu <i>Case folding</i> , <i>tokenizing</i> , <i>stopword remover</i> , dan <i>stemming</i> .
6	Ivan Boban, dkk	<i>"Sentence retrieval using Stemming and Lemmatization with Different Length of the Queries"</i>	1. TS-ISF 2. <i>Stop word removal</i> 3. <i>Stemming</i> 4. <i>Lemmatization</i>	Hasil dari penelitian menunjukkan bahwa <i>lemmatization</i> menunjukkan hasil yang lebih baik ketika digunakan pada <i>query</i> yang lebih panjang dibandingkan <i>stemming</i> dengan hasil akhir 4 dari 6 hasil tes menunjukkan nilai P@10, MAP, dan R-prec lebih baik menggunakan <i>lemmatization</i> dibandingkan <i>stemming</i> yang	Penggunaan metode <i>stemming</i> , <i>stopword removal</i> dan <i>lemmatization</i> .

No	Penulis	Judul	Metode	Hasil	Persamaan
				hanya memiliki 2 hasil tes yang lebih baik.	

Sebelumnya telah dilakukan penelitian serupa berjudul “*Automated Essay Scoring* menggunakan *Cosine Similarity* pada Penilaian Esai Multi Soal” oleh Alfirna Rizqi Lahitani pada tahun 2021 [3]. Penelitian ini bertujuan untuk menerapkan bidang ilmu *Automated Essay Scoring* (AES) agar bisa memudahkan proses pengoreksian dan pemberian skor jawaban esai dibandingkan skoring manual yang membutuhkan waktu panjang. Penelitian ini menerapkan metode pembobotan TF-IDF dan pengukuran kemiripan menggunakan cosine similarity pada jawaban esai dokumen siswa dan dokumen ahli sebelum melakukan skoring. Hasil penelitian menunjukkan bahwa Nilai akhir yang diperoleh sangat kecil untuk sebuah skor yang proporsional karena jika rata-rata pemberian nilai akhir berkisar antara 80-100, maka *cosine similarity* mewakili 52% poin dibandingkan dengan nilai aslinya dan tidak memiliki unsur subjektif. Keterbatasan dari penelitian adalah tidak diimplementasikannya metode evaluasi untuk mengukur tingkat akurasi dari model yang telah dibuat.

Masih bertemakan penilaian esai, terdapat penelitian berjudul “Penilaian Esai Pendek Otomatis dengan Pencocokan Kata Kunci Frasa Nomina” yang dilakukan oleh Nurul Chamidah, dkk pada tahun 2021 [2]. Penelitian dilakukan untuk mengatasi masalah terkait dengan inkonsistensi dan alokasi waktu yang besar bila melakukan penilaian esai secara manual. Penelitian ini dilakukan dengan mengekstrak kata kunci dan menilainya berdasarkan kecocokan. Beberapa tahap yang dilakukan pada dokumen uji dan ahli yaitu *case folding*, *POS Tagging*, ekstraksi frasa, dan *stemming*. Hasil evaluasi menunjukkan *Mean Absolute Error* (MAE) sebesar 18% dan *Pearson Correlation* sebesar 0.83. Salah satu keunggulan dari penelitian ini yaitu digunakannya metode *POS Tagging* yang dapat mengelompokkan kata berdasarkan kelasnya.

Terdapat penelitian serupa berjudul “Pencocokan Berbasis Kata Kunci pada Penilaian Esai Pendek Otomatis Berbahasa Indonesia” oleh Nurul Chamidah, dkk pada tahun 2021 [5]. Penelitian ini bertujuan untuk membuat evaluator penilai esai otomatis agar penilaian lebih konsisten. Penelitian dilakukan dengan menghitung kecocokan antara jawaban uji dan jawaban referensi dengan melihat kata kunci masing-masing jawaban. Hasil menunjukkan nilai *Mean Absolute Error* (MAE) sebesar 0.25 dan

Pearson Correlation 0.79. Kekurangan penelitian yaitu nilai MAE dari sistem masih tergolong besar dan mempunyai perbedaan cukup signifikan jika dibandingkan dengan penilaian evaluator.

Selanjutnya, terdapat penelitian berjudul “Penerapan *Cosine Similarity* dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen” oleh Ade Riyani, dkk di tahun 2019 [7]. Penelitian ini bertujuan untuk mendeteksi kemiripan dokumen sebagai cara untuk menanggulangi dan mencegah kasus plagiarisme di dunia penulisan. Penelitian ini menggunakan metode pembobotan TF-IDF dan perhitungan kemiripan dengan *cosine similarity*. Hasil penelitian menunjukkan *cosine similarity* dan pembobotan TF-IDF telah berhasil mendeteksi kemiripan suatu dokumen dan proses *stemming* sangat berpengaruh pada nilai kemiripan hasil yakni menjadi lebih tinggi. Nilai rata-rata perbedaan nilai kemiripan jika menggunakan *stemming* dan tidak menggunakan *stemming* sebesar 10%. Keunggulan dari penelitian ini yaitu dibahasnya perbandingan penelitian ketika menggunakan dan tidak menggunakan proses *stemming* pada tahap *preprocessing* yang ternyata sangat berpengaruh pada hasil nilai kemiripan.

Lalu, terdapat penelitian berjudul “*Detector Similarity Answers Between Students on Essay Digital Exam System*” oleh Regi Ismayana Pratama, dkk pada tahun 2019 [8]. Penelitian ini bertujuan untuk mengatasi permasalahan peserta didik yang mencontek saat ujian. Penelitian ini menggunakan metode *term frequency*, n-gram, dan *cosine similarity* untuk penilaian esai dan kemiripannya. Murid dengan tingkat kemiripan 75% atau lebih akan dianggap mencontek. Hasil penelitian menunjukkan 90% dari rekomendasi sistem sesuai dengan penilaian guru pada jawaban menyontek. Sistem mampu mendukung guru dalam mengevaluasi ujian atau belajar. N-gram dan *cosine similarity* merupakan metode yang baik untuk mendeteksi kemiripan jawaban murid.

Terakhir, terdapat penelitian berjudul “*Sentence retrieval using Stemming and Lemmatization with Different Length of the Queries*” oleh Ivan Boban, Alen Doko, dan Sven Gotovac pada tahun 2020 [11]. Penelitian ini bertujuan untuk membandingkan penggunaan *stemming* dan *lemmatization* pada *sentence retrieval*. Beberapa metode yang digunakan antara lain TS-ISF, *Stopword removal*, *Stemming*, dan *Lemmatization*. Hasil dari penelitian menunjukkan bahwa *lemmatization*

menunjukkan hasil yang lebih baik ketika digunakan pada *query* yang lebih panjang dibandingkan *stemming* dengan hasil akhir 4 dari 6 hasil tes menunjukkan nilai P@10, MAP, dan R-prec lebih baik menggunakan *lemmatization* dibandingkan *stemming* yang hanya memiliki 2 hasil tes yang lebih baik.

Berdasarkan enam penelitian di atas, terdapat beberapa perbedaan antara penelitian yang telah dilakukan dan penelitian yang akan dikembangkan penulis yaitu pada jenis teks esai pendek yang digunakan dimana peneliti akan menggunakan teks berbahasa Inggris. Artinya, *library* dan algoritma yang digunakan tentu akan sangat berbeda dengan penelitian sebelumnya yang menggunakan teks esai berbahasa Indonesia. Objek penelitian yang akan dilakukan berfokus pada penilaian esai pendek dengan metode pembobotan TF-IDF dan pengukuran similaritas dengan *cosine similarity*, berbeda dengan penelitian sebelumnya yang menggunakan pencocokan kata kunci, *pearson correlation*, n-gram, TS-ISF dan sebagainya.

Selain itu, terdapat penerapan dan perbandingan antara penggunaan *lemmatization* dan *stemming* pada penelitian ini. *Lemmatization* merupakan teknik modifikasi dari *stemming*. *Stemming* menghasilkan kata ke bentuk awal (*root*) namun belum tentu kata tersebut memiliki makna. Sedangkan *lemmatization* memastikan kata dasar yang dihasilkan merupakan kata yang memiliki makna [10] Terakhir, penelitian ini akan menggunakan evaluasi untuk mengukur tingkat akurasi model menggunakan perhitungan *Mean Absolute Error* (MAE).

2.2 Dasar Teori

Terdapat beberapa teori yang digunakan dalam penelitian yang dikerjakan antara lain ujian esai singkat, *text mining*, *natural language processing*, *preprocessing*, *case folding*, *POS tagging*, *stopword removal*, *stemming*, *lemmatization*, *tokenizing*, TF-IDF, *cosine similarity*, dan *mean absolute error* (MAE).

2.2.1 Ujian Esai Singkat

Evaluasi dan penilaian dibutuhkan dalam proses pembelajaran guna mengukur pemahaman peserta didik serta kualitas dari pembelajaran itu sendiri [12]. Ujian yang diberikan dapat berbentuk soal objektif atau soal esai singkat. Jawaban dari soal esai singkat umumnya berupa uraian esai singkat yang terdiri dari 1 frasa (dua kata atau

lebih) hingga 1 paragraf atau paling banyak 100 kata [13]. Dalam mengukur kedalaman pemahaman peserta didik soal esai singkat dinilai lebih cocok untuk diterapkan karena dapat mengatasi kelemahan daya ukur dari soal objektif yang hanya berupa pilihan ganda [14].

Melalui evaluasi soal esai singkat, peserta didik dilatih untuk menjawab soal dengan menjelaskan ide, pendapat, dan alasan menggunakan nalar dalam bentuk kata dan bahasa mereka sendiri [15]. Soal esai singkat biasanya digunakan untuk meningkatkan kemampuan *problem-solving*, *high-order thinking*, dan pengetahuan peserta didik [16]. Namun, hal tersebut menyebabkan banyaknya variasi jawaban yang dimiliki peserta didik sehingga proses evaluasi dan penilaian jawaban akan lebih lama, subjektivitas yang lebih tinggi, kompleks, dan inkonsisten [17]. Hasil evaluasi yang inkonsisten dapat dianggap tidak adil bagi peserta didik [15]. Oleh karena itu, dibutuhkan evaluator otomatis agar proses penilaian jawaban dapat dilakukan lebih mudah, cepat, dan konsisten.

2.2.2 Penilaian Esai Otomatis

Sistem penilaian esai otomatis merupakan sistem yang bekerja dengan cara membandingkan dan mencocokkan dokumen jawaban peserta didik dan dokumen ahli [4]. Sistem bekerja menyerupai cara seorang penilai melakukan evaluasi penilaian jawaban esai [18]. Sistem ini akan memberikan skor kemiripan antara jawaban esai peserta didik dan kunci jawaban dari dokumen ahli. Semakin mirip skor yang dihasilkan dokumen jawaban dan dokumen ahli maka semakin baik hasil penilaiannya.

2.2.3 Text Mining

Text mining merupakan cabang ilmu dari *data mining*. *Text mining* merupakan proses analisa teks yang dilakukan untuk mendapatkan informasi berguna. Perbedaannya terletak pada bentuk datanya dimana data mining memiliki bentuk data terstruktur dan data *text mining* tidak terstruktur [19]. Fungsi *text mining* meliputi *categorization*, *clustering*, *entity extraction*, *sentiment analysis*, dan *summarization* [20]. Beberapa penelitian yang menggunakan *text mining* beragam mulai dari *word matching*, deteksi plagiarisme, *automated essay scoring*, *sentiment analysis* dan *topic modelling*, peringkasan teks berita dan sebagainya. Dalam prosesnya, biasanya

digunakan teknik *Natural Language Processing* (NLP) atau Pengolahan Bahasa Alami seperti *stemming*, *POS tagging*, *lemmatization*, dan lain-lain.

2.2.4 Natural Language Processing (NLP)

Natural Language Processing (NLP) atau Pengolahan Bahasa Alami merupakan cabang dari ilmu *Artificial Intelligence* (AI) atau kecerdasan buatan. NLP membantu komputer untuk mengerti, memanipulasi, dan menginterpretasikan bahasa manusia [21]. Terdapat 2 teknik NLP yaitu analisis sintaksis dan analisis semantik. Analisis sintaksis menganalisa teks dengan prinsip tata bahasa (*grammar*) dasar untuk mendeteksi struktur kalimat, mengatur kata-kata dan menyambungkannya. Sedangkan analisis semantik berfungsi untuk menentukan makna ambigu yang digunakan pada konteks tertentu. Beberapa teknik yang digunakan pada NLP antara lain *stemming*, *lemmatization*, *POS tagging*, *keyword extraction*, *summarization* dan sebagainya.

2.2.5 Preprocessing

Tahap *preprocessing* merupakan tahap pertama yang dilakukan dalam proses *text mining* [22] Tahap ini berisikan proses membersihkan dan menyiapkan data teks sebelum diproses analisis [9]. Pada tahap ini, bentuk data yang awalnya tidak terstruktur akan diubah menjadi data terstruktur sesuai kebutuhan proses analisis di tahap selanjutnya [4]. Beberapa tahap *text preprocessing* penelitian menggunakan teknik yaitu *case folding*, *stopword removal*, *stemming*, *lemmatization*, dan *tokenizing*.

2.2.6 Case Folding

Proses ini diperlukan untuk mengubah semua huruf di dalam sebuah teks dokumen menjadi bentuk standar (huruf kecil) [4]. Fungsinya adalah untuk mengidentifikasi kata yang sebenarnya sama walau ditulis dengan huruf kapital. Sebagai contoh untuk mengidentifikasi bahwa kata “*Case*” dan “*CASE*” adalah kata yang sama, maka kedua kata tersebut akan diubah menjadi huruf kecil yaitu “*case*”.

2.2.7 Tokenizing

Tokenizing merupakan proses yang dilakukan untuk memotong *string input* dari teks dokumen berdasarkan setiap kata yang menyusunnya [4]. Spasi dan tanda

baca digunakan untuk mengidentifikasi dan memisahkan kata pada sebuah kalimat. Sebagai contoh terdapat sebuah kalimat “*Direct trade process*” yang akan menghasilkan 5 token yaitu “*process*”, “*sell*”, “*buy*”, “*in*”, dan “*direct*” [19]. Untuk melakukan proses ini, dapat digunakan *library* NLTK.

2.2.8 POS Tagging

POS *Tagging* merupakan salah satu proses di dalam NLP yang berfungsi untuk memberikan *parts of speech* pada kata. *Part of speech* melingkupi *nouns*, *verbs*, *adverbs*, *adjectives*, *pronouns*, *conjunctions* dan sub-kategorinya. Proses ini menggunakan beberapa data seperti referensi kata, *vocabularies*, aturan dan lain sebagainya. Proses ini menfaatkan informasi probabilistik untuk mengatasi ketidakjelasan kategori sebuah kata seperti contohnya kata “*escape*” bisa menjadi *verb* dan bisa juga *noun* [23].

POS *Tagging* pada *library* NLTK memanfaatkan set label dari *Penn Treebank*. Berikut merupakan set label POS pada *Penn Treebank*.

Tabel 2.2 POS Tags Penn Treebank Project

Tag	Description	Tag	Description
CC	Coordinating conjunction	PRP\$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	<i>to</i>
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass	VBN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non-3rd person singular present
NNP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NNPS	Proper noun, plural	WDT	Wh-determiner

Tag	Description	Tag	Description
PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP\$	Possessive wh-pronoun
PRP	Personal pronoun	WRB	Wh-adverb

2.2.9 Stopword Removal

Dalam penulisan bahasa Inggris, terdapat kata *grammatical* dan kata *lexical*. Kata *grammatical* meliputi artikel, kata ganti (*pronoun*), dan konjungsi sedangkan kata *lexical* meliputi kata benda (*noun*), kata kerja (*verb*), dan kata sifat (*adjective*). Kata *grammatical* tidak berkontribusi pada makna kalimat dan hanya berfungsi untuk melengkapi makna logis [24]. Kata seperti inilah yang disebut sebagai *stopword*. *Stopword* seharusnya dihapus dari sebuah kalimat sebelum teknik *natural language processing* diterapkan. Proses inilah yang disebut sebagai *stopword removal*. Tujuannya adalah untuk menghilangkan kata-kata yang sangat umum dan dianggap tidak terlalu bernilai untuk proses analisis teks dokumen.

2.2.10 Stemming

Stemming adalah proses yang menghasilkan bentuk dasar (*root/stem*) dari sebuah kata [25]. Proses ini dilakukan dengan memisahkan sebuah kata dari kata dasar dan imbuhan seperti awalan (*prefix*) dan akhiran (*suffix*) lalu menghapus imbuhan [5]. Tujuannya adalah ketika melakukan perbandingan dokumen jawaban peserta didik dan dokumen ahli, kata yang dibandingkan berbentuk kata dasar dan tidak dipengaruhi oleh bentuk kata aktif, pasif, pembendaan, sifat, dan lain sebagainya. Sebagai contoh dalam bahasa Inggris terdapat *suffix* seperti “-ing” untuk *present continuous tense* dan “-ed” untuk *past tense*. Jika dalam teks dokumen terdapat kata “cooked” dan “cooking” yang pada sistem ingin dikenali sebagai kata “cook” karena pada dasarnya ketiga kata tersebut memiliki arti yang sama yaitu “memasak” dan hanya berbeda *tenses*, maka proses *stemming* akan berguna disini. Beberapa contoh algoritma *stemming* antara lain *Porter stemmer*, *Snowball stemmer*, *Lancaster stemmer*, dan *Regular Expression stemmer*.

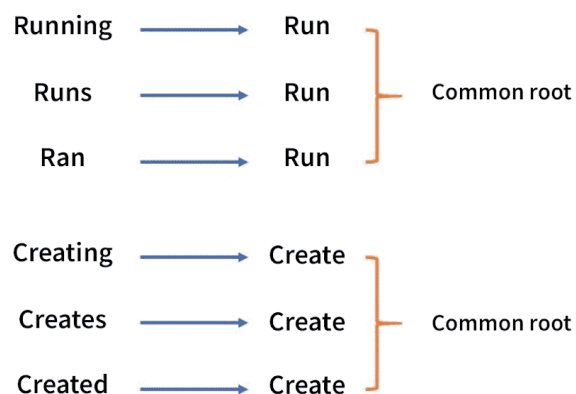
Salah satu algoritma *stemmer* yang umum digunakan dan dianggap memiliki performa yang baik adalah *Snowball stemmer*. *Snowball* atau kadang juga disebut sebagai *Porter2* merupakan algoritma *Porter* yang telah dimodifikasi dan ditingkatkan

[26]. Algoritma ini memiliki tingkat akurasi yang lebih tinggi dibanding algoritma aslinya. Algoritma *Porter* sendiri merupakan algoritma *stemmer* paling banyak digunakan karena terbukti efektif dan mudah digunakan [27]. Beberapa perubahannya antara lain :

1. Akhiran 'y' berubah menjadi 'i' jarang terjadi.
2. Sufiks 'us' tidak kehilangan 's' nya.
3. Penghapusan akhiran tambahan, termasuk akhiran 'ly'.
4. Tambahkan langkah 0 untuk menangani apostrof (Juli 2005).
5. Daftar kecil bentuk-bentuk luar biasa disertakan (Nov 2006).

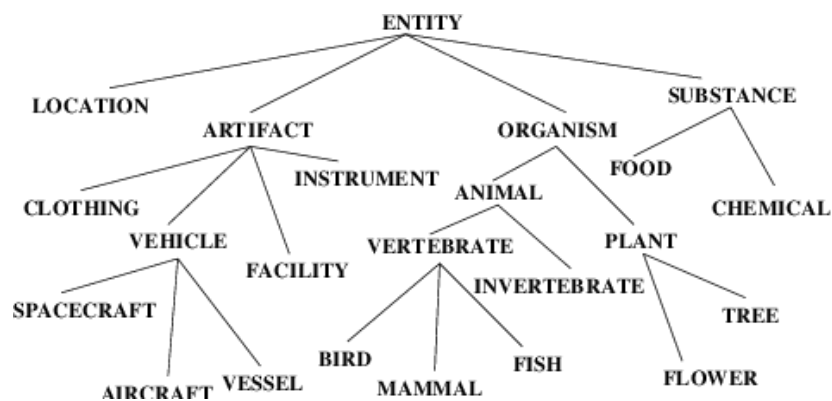
2.2.11 Lemmatization

Lemmatization merupakan modifikasi dari *stemming* dan bersifat lebih kuat karena melakukan analisis morfologis dari kata yang diproses sebagai pertimbangan. Proses ini mengubah kata ke bentuk dasar dan menjadikannya kata yang bermakna sesuai konteks penggunaannya [28]. Berbeda dengan *stemming* yang hanya menghilangkan awalan dan akhiran kata, *lemmatization* akan mengubah kata sesuai dengan bentuk kamusnya. Sebagai contoh kata “*leafs*” dan “*leaves*” jika menggunakan *stemming* akan berubah menjadi “*leaf*” dan “*leav*” dimana “*leav*” tidak memiliki arti. Namun, jika menggunakan *lemmatization* maka “*leafs*” dan “*leaves*” akan berubah menjadi “*leaf*”. Penggunaan *lemmatization* dapat melalui *library* NLTK yang menyediakan kelas *WordNetLemmatizer* [25]. Berikut merupakan ilustrasi proses *lemmatization*.



Gambar 2.1 Proses Lematisasi

Salah satu algoritma lematisasi yang umum digunakan dan dianggap memiliki performa yang baik adalah *WordNet*. *WordNet* merupakan basis data linguistik yang luas, terbuka, dan bebas untuk bahasa Inggris yang menyediakan asosiasi tata bahasa yang sistematis antara kata-kata. *WordNet* menyediakan fasilitas *lemmatization* dan merupakan salah satu *lemmatizer* yang paling sering digunakan dan paling orisinil [29]. *WordNet* akan mengumpulkan kata-kata berdasarkan maknanya dengan cara menemukan makna dari kata dan menghilangkan ambiguitasnya dengan menambahkan konteks lalu mengklasifikasikan relasi sintaksis antar kata dengan urutan tertentu sebagai *vocabulary* [29]. Berikut merupakan gambaran hierarki kata-kata pada *WordNet*.



Gambar 2.2 Hierarki *WordNet*

2.2.12 TF-IDF

TF-IDF merupakan metode yang menggabungkan 2 konsep perhitungan bobot yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). Dalam prosenya, dilakukan perhitungan jumlah kemunculan (frekuensi) dari term di setiap dokumennya. Hal inilah yang disebut sebagai *Term Frequency* (TF). Semakin tinggi nilai TF dalam sebuah dokumen, maka semakin besar bobotnya atau nilai kesesuaiannya [4]. Sedangkan IDF melakukan perhitungan seberapa luas *term* didistribusikan dalam kumpulan dokumen yang berkaitan. IDF akan menunjukkan hubungan keberadaan *term* di seluruh dokumen. Semakin sedikit jumlah dokumen yang mengandung *term* tersebut, maka semakin besar nilai IDF-nya [4]. Untuk mendapatkan nilai IDF digunakan formulasi sebagai berikut.

$$idf_t = \log \frac{N}{df_t + 1}$$

(Rumus 2. 1)

Keterangan :

idf_t : *inverse document frequency*

N : jumlah dokumen

df_t : *document frequency*

Dengan idf_t merupakan IDF dihitung pada seluruh term yang ada, N merupakan jumlah keseluruhan dokumen, df_t merupakan total keberadaan *term* pada seluruh dokumen. Selanjutnya, untuk mendapatkan bobot TF-IDF maka digunakan formulasi sebagai berikut.

$$tfidf(t, d) = tf(t, d) \cdot idf(t)$$

(Rumus 2. 2)

Keterangan :

$tfidf(t, d, D)$: bobot TF-IDF pada masing-masing *term*

$tf(t, d)$: *term frequency* pada masing-masing dokumen

$idf(t, D)$: nilai *inverse document frequency* dari sebuah term

2.2.13 Cosine Similarity

Cosine similarity merupakan nilai perhitungan *cosinus* sudut yang dipakai untuk mengukur kemiripan antara dua buah vektor pada sebuah ruang dimensi [30]. Setiap vektor merepresentasikan setiap kata dalam setiap teks dokumen yang dibandingkan membentuk segitiga. Kemudian, diterapkan hukum cosinus untuk menyatakannya. Jika dua buah dokumen identik maka sudutnya akan bernilai 0 derajat dan kesamaannya bernilai 1. Sebaliknya, jika dua dokumen tidak identik maka sudutnya akan bernilai 90 derajat dan kesamaannya bernilai 0 [4]. Rumus perhitungannya adalah sebagai berikut.

$$\cos a = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

(Rumus 2. 3)

Keterangan :

A : vektor A

B : vektor B

|A| : panjang vektor A

|B| : panjang vektor B

2.2.14 Mean Absoulte Error (MAE)

Perhitungan *Mean Absoulte Error* (MAE) dilakukan untuk mengukur performa model yang telah dibangun. MAE termasuk ke dalam jenis *statistical accuracy metrics* yang paling umum digunakan dalam perhitungan akurasi dari *recommender system* [31]. Nilainya didapatkan dengan menghitung rata-rata kesalahan antara nilai yang diperoleh oleh model dengan nilai yang diberikan oleh penilai (manusia) [2]. Semakin kecil nilainya, maka semakin akurat penilaian model. Rumus perhitungannya adalah sebagai berikut.

$$MAE = \frac{\sum |n_{uji} - n_{model}|}{Jum}$$

(Rumus 2. 4)

Keterangan :

n_{sistem} : nilai model

n_{uji} : nilai evaluator

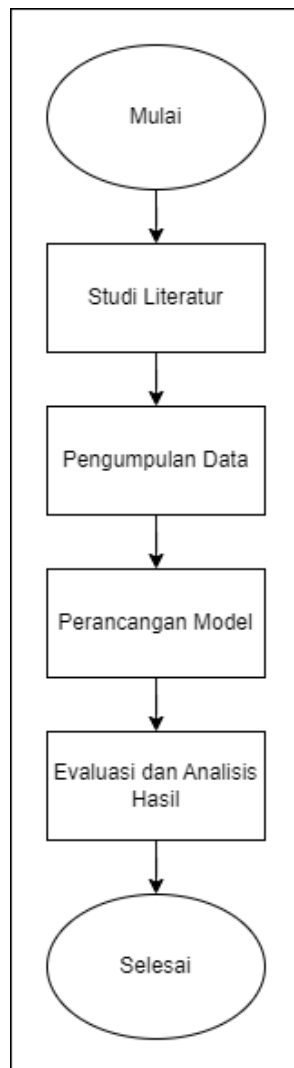
Jum : jumlah seluruh jawaban peserta didik

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Alur penelitian dapat dilihat pada Gambar 3.1 Alur Penelitian. Alur dimulai dari studi literatur, pengumpulan data, perancangan model, dan analisis hasil. Pada tahap ketiga yaitu Perancangan Model, terdapat alur seperti yang dapat dilihat pada Gambar 3.2 Alur Perancangan Model. Setelah semua tahapan tersebut selesai, pada analisis asil akan dilakukan tahap evaluasi dengan membandingkan nilai model dan nilai hasil evaluator manusia. Terakhir, akan dilakukan analisis dan perbandingan antara tingkat akurasi proses *stemming* dan *lemmatization*.



Gambar 3.1 Alur Penelitian

3.2 Penjabaran Langkah Penelitian

Berikut merupakan penjabaran dari setiap langkah penelitian yang telah digambarkan pada gambar diagram 3.1.

3.2.1 Studi Literatur

Perancangan pemodean penilaian esai singkat ini memerlukan pemahaman teoritis terhadap metode dan proses yang diterapkan dalam pemodelannya dari berbagai referensi jurnal dan buku. Hal ini dilakukan agar persiapan dan pemrosesan data dilaksanakan dengan tahapan dan fungsi yang tepat. Dengan begitu, diharapkan pada pelaksanaannya pemrosesan data dapat dilakukan dengan efektif dan efisien serta memiliki hasil akurasi yang baik.

3.2.2 Pengumpulan Data

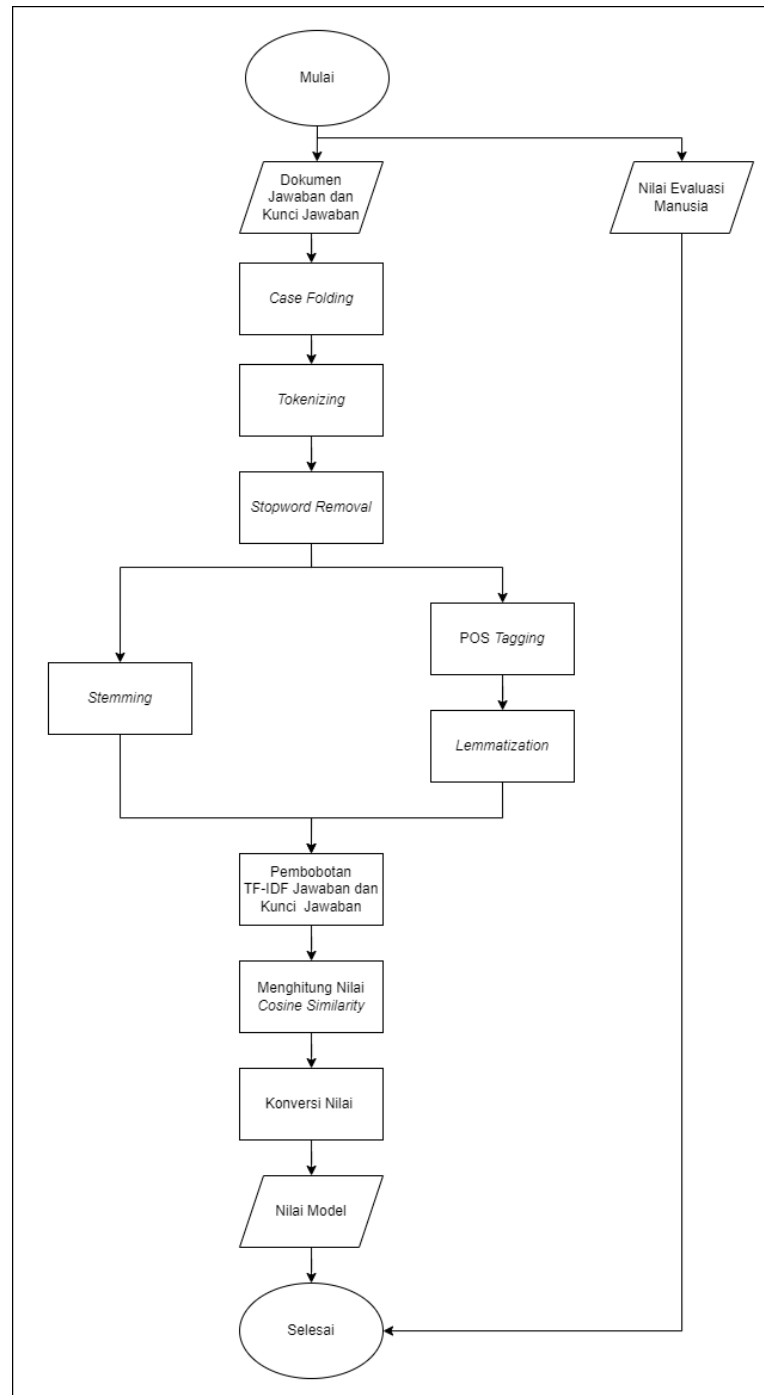
Data yang akan digunakan untuk pemrosesan perlu dikumpulkan dan disiapkan terlebih dahulu. Data yang akan digunakan terdiri dari 2 jenis data yaitu dokumen jawaban milik peserta didik dan dokumen jawaban ahli sebagai pembandingnya. Selain data berupa dokumen jawaban, akan disiapkan data berupa hasil penilaian yang dilakukan oleh evaluator manusia. Hasil penilaian ini akan digunakan sebagai pembanding untuk menentukan tingkat akurasi penilaian model.

3.2.3 Perancangan Model

Setelah melewati berbagai tahap persiapan, tahap perancangan model dapat dilakukan. Terdapat beberapa tahap di dalam perancangan model yaitu *text pre-processing* berupa *case folding*, *tokenizing*, *stopword removal*, *POS tagging*, *stemming* dan *lemmatization*, lalu pembobotan TF-IDF untuk kedua dokumen pada proses *stemming* dan *lemmatization*, perhitungan tingkat similaritas dengan *cosine similarity*, dan konversi nilai menggunakan. Beberapa tahapannya dijelaskan lebih lanjut pada gambar Gambar 3.2 Alur Perancangan Model.

Di dalam tahap perancangan model, alur dimulai dari pemrosesan data yang terdiri dari data teks jawaban peserta didik, teks kunci jawaban ahli, dan nilai hasil evaluasi dari evaluator manusia. Data jawaban peserta didik dan data kunci jawaban ahli akan melalui tahapan *text pre-processing* yang terdiri dari *case folding*, *tokenizing*,

stopword removal, *POS tagging*, *stemming* dan *lemmatization*. Hasil dari tahap *pre-processing* akan dibobotkan menggunakan metode pembobotan TF-IDF. Hasil nilai pembobotan akan digunakan untuk menghitung nilai kemiripan antara jawaban peserta didik dan kunci jawaban ahli menggunakan metode *cosine similarity*. Selanjutnya, nilai *cosine similarity* akan dikonversi dan didapatkan hasil perhitungan nilai model.



Gambar 3.2 Alur Perancangan Model

3.2.3.1 Case Folding

Sebelum melakukan *text pre-processing*, diperlukan *import library* yang akan dibutuhkan. Dalam penelitian ini, akan digunakan library NLTK (Natural Language Toolkit) dalam pemrosesan menggunakan Natural Language Processing (NLP). Selanjutnya, akan dilakukan *import data* untuk kemudian diproses ke beberapa tahap antara lain *case folding*, *tokenizing*, *stopword removal*, *POS tagging*, *stemming* dan *lemmatization*. Berikut adalah contoh pertanyaan dan dokumen yang akan melalui tahapan *text pre-processing*.

Pertanyaan : “Where are variables declared in a C++ program?”

Tabel 3.1 Jawaban Peserta Didik

<i>Local variables are better declared inside the method or function they will be used in at the beginning of the method. Global variables are declared outside of any function or method, generally before functions are defined.</i>
--

Tabel 3.2 Jawaban Ahli

<i>Variables can be well declared anywhere in a program. They can be declared inside a function (local variables) or outside the functions (global variables)</i>

Pada tahap ini, dokumen teks akan melalui proses perubahan huruf kapital menjadi huruf kecil dengan tujuan kata-kata yang sama terdeteksi sebagai kata yang sama, tidak terdeteksi berbeda karena perbedaan huruf kapital di dalam katanya. Selain itu, karakter selain huruf abjad seperti tanda baca dan angka akan dihilangkan. Berikut merupakan hasil sebelum dan sesudah tahap *case folding*.

Tabel 3.3 Case Folding Jawaban Peserta Didik

Sebelum	Sesudah
<i>Local variables are better declared inside the method or function they will be used in at the beginning of the method. Global variables are declared outside of any function or method, generally before functions are defined.</i>	<i>local variables are better declared inside the method or function they will be used in at the beginning of the method global variables are declared outside of any function or method generally before functions are defined after declaration of the variables data type</i>

Tabel 3.4 Case Folding Jawaban Ahli

Sebelum	Sesudah
<i>Variables can be well declared anywhere in a program. They can be declared inside a function (local variables) or outside the functions (global variables)</i>	<i>variables can be well declared anywhere in a program they can be declared inside a function local variables or outside the functions global variables</i>

3.2.3.2 Tokenizing

Setelah semua huruf pada teks menjadi huruf kecil dan tidak memiliki tanda baca, simbol, atau angka maka akan dilakukan proses *tokenizing*. Proses ini berfungsi untuk memisahkan atau memecahkan kalimat menjadi kata-kata yang menyusunnya. Berikut merupakan hasil sebelum dan sesudah tahap *tokenizing*.

Tabel 3.5 Tokenizing Jawaban Peserta Didik

Sebelum	Sesudah
<i>local variables are better declared inside the method or function they will be used in at the beginning of the method global variables are declared outside of any function or method generally before functions are defined after declaration of the variables data type</i>	<i>['local', 'variables', 'are', 'better', 'declared', 'inside', 'the', 'method', 'or', 'function', 'they', 'will', 'be', 'used', 'in', 'at', 'the', 'beginning', 'of', 'the', 'method', 'global', 'variables', 'are', 'declared', 'outside', 'of', 'any', 'function', 'or', 'method', 'generally', 'before', 'functions', 'are', 'defined']</i>

Tabel 3.6 Tokenizing Jawaban Ahli

Sebelum	Sesudah
<i>variables can be well declared anywhere in a program they can be declared inside a function local variables or outside the functions global variables</i>	<i>['variables', 'can', 'be', 'well', 'declared', 'anywhere', 'in', 'a', 'program', 'they', 'can', 'be', 'declared', 'inside', 'a', 'function', 'local', 'variables', 'or', 'outside', 'the', 'functions', 'global', 'variables']</i>

3.2.3.3 Stopword Removal

Stopword adalah kata-kata umum di dalam teks yang biasanya tidak memiliki makna dan muncul dalam frekuensi yang banyak. Penghilangan kata-kata ini tidak akan mempengaruhi makna kalimat dan dapat meningkatkan keakuratan hasil analisis

yang akan dilakukan. Proses penghapusan inilah yang disebut sebagai *stopword removal*. Beberapa contoh *stopword* dalam bahasa Inggris antara lain “are”, “i”, “am”, “the”, “was”, “a”, “to” dan sebagainya. Berikut merupakan hasil sebelum dan sesudah tahap *stopword removal*.

Tabel 3.7 *Stopword Removal* Jawaban Peserta Didik

Sebelum	Sesudah
['local', 'variables', 'are', 'better', 'declared', 'inside', 'the', 'method', 'or', 'function', 'they', 'will', 'be', 'used', 'in', 'at', 'the', 'beginning', 'of', 'the', 'method', 'global', 'variables', 'are', 'declared', 'outside', 'of', 'any', 'function', 'or', 'method', 'generally', 'before', 'functions', 'are', 'defined']	['local', 'variables', 'better', 'declared', 'inside', 'method', 'function', 'used', 'beginning', 'method', 'global', 'variables', 'declared', 'outside', 'function', 'method', 'generally', 'functions', 'defined']

Tabel 3.8 *Stopword Removal* Jawaban Ahli

Sebelum	Sesudah
['variables', 'can', 'be', 'well', 'declared', 'anywhere', 'in', 'a', 'program', 'they', 'can', 'be', 'declared', 'inside', 'a', 'function', 'local', 'variables', 'or', 'outside', 'the', 'functions', 'global', 'variables']	['variables', 'well', 'declared', 'anywhere', 'program', 'declared', 'inside', 'function', 'local', 'variables', 'outside', 'functions', 'global', 'variables']

3.2.3.4 Stemming

Pada bahasa Inggris, penggunaan sebuah kata memiliki variasi seperti kata aktif dan kata pasif. Variasi kata ini memiliki akar atau dasar kata yang membentuknya. Oleh karenanya, variasi ini perlu dihilangkan dan dikembalikan ke bentuk akarnya agar hasil analisis yang akan dilakukan menjadi lebih baik dan akurat. Proses inilah yang disebut sebagai *stemming*. Berikut merupakan hasil sebelum dan sesudah tahap *stemming*.

Tabel 3.9 *Stemming* Jawaban Peserta Didik

Sebelum	Sesudah
['local', 'variables', 'better', 'declared', 'inside', 'method', 'function', 'used', 'beginning']	['local', 'variabl', 'better', 'declar', 'insid', 'method', 'function', 'use', 'begin', 'method', 'global', 'variabl']

Sebelum	Sesudah
'method', 'global', 'variables', 'declared', 'outside', 'function', 'method', 'generally', 'functions', 'defined']	'declar', 'outsid', 'function', 'method', 'general', 'function', 'defin']

Tabel 3.10 *Stemming* Jawaban Ahli

Sebelum	Sesudah
['variables', 'well', 'declared', 'anywhere', 'program', 'declared', 'inside', 'function', 'local', 'variables', 'outside', 'functions', 'global', 'variables']	[variabl', 'well', 'declar', 'anywhere', 'program', 'declar', 'insid', 'functionlocal', 'variabl', 'outsid', 'function', 'global', 'variabl']

3.2.3.5 POS Tagging

POS *Tagging* merupakan proses yang berfungsi untuk memberikan *parts of speech tag* pada kata. *Part of speech* melingkupi *nouns, verbs, adverbs, adjectives, pronouns, conjunctions* dan sub-kategorinya. Kumpulan kata yang telah diproses dan di-filter akan diberikan *tag* sesuai dengan kategorinya untuk meningkatkan akurasi pada proses *lemmatization* nantinya. Berikut merupakan hasil sebelum dan sesudah tahap *POS tagging*.

Tabel 3.11 *POS Tagging* Jawaban Peserta Didik

Sebelum	Sesudah
['local', 'variables', 'better', 'declared', 'inside', 'method', 'function', 'used', 'beginning', 'method', 'global', 'variables', 'declared', 'outside', 'function', 'method', 'generally', 'functions', 'defined']	[('local', 'JJ'), ('variables', 'NNS'), (('better', 'RB'), ('declared', 'VBD'), (('inside', 'JJ'), ('method', 'NN'), (('function', 'NN'), ('used', 'VBN'), (('beginning', 'VBG'), ('method', 'NN'), ('global', 'JJ'), ('variables', 'NNS'), ('declared', 'VBD'), ('outside', 'JJ'), ('function', 'NN'), ('method', 'NN'), ('generally', 'RB'), ('functions', 'NNS'), ('defined', 'VBD'))]

Tabel 3.12 *POS Tagging* Jawaban Ahli

Sebelum	Sesudah
['variables', 'well', 'declared', 'anywhere', 'program', 'declared', 'inside', 'function', 'local', 'variables', 'defined']	[('variables', 'NNS'), ('well', 'RB'), (('declared', 'VBN'), ('anywhere', 'RB'), ('program', 'NN'), ('declared', 'VBN'), ('inside', 'IN'), ('function', 'NN'), ('local', 'IN'), ('variables', 'NNS'), ('defined', 'VBD'))]

Sebelum	Sesudah
'outside', 'functions', 'global', 'variables']	'NN'), ('local', 'JJ'), ('variables', 'NNS'), ('outside', 'IN'), ('functions', 'NNS'), ('global', 'JJ'), ('variables', 'NNS')]

3.2.3.6 Lemmatization

Lemmatization memiliki fungsi yang sama dengan *stemming*, yakni menghasilkan kata ke bentuk akarnya. Namun, bedanya *lemmatization* memastikan kata yang diubah menjadi bentuk akar tersebut valid dan memiliki makna sesuai seperti di kamus dibandingkan *stemming* yang hanya menghapus *prefix* dan *suffix* dari sebuah kata. Berikut merupakan hasil sebelum dan sesudah tahap *lemmatization*

Tabel 3.13 *Lemmatization* Jawaban Peserta Didik

Sebelum	Sesudah
[('local', 'JJ'), ('variables', 'NNS'), ('better', 'RB'), ('declared', 'VBD'), ('inside', 'JJ'), ('method', 'NN'), ('function', 'NN'), ('used', 'VBN'), ('beginning', 'VBG'), ('method', 'NN'), ('global', 'JJ'), ('variables', 'NNS'), ('declared', 'VBD'), ('outside', 'JJ'), ('function', 'NN'), ('method', 'NN'), ('generally', 'RB'), ('functions', 'NNS'), ('defined', 'VBD')]	[('local', 'variable', 'well', 'declare', 'inside', 'method', 'function', 'use', 'begin', 'method', 'global', 'variable', 'declare', 'outside', 'function', 'method', 'generally', 'function', 'define']

Tabel 3.14 *Lemmatization* Jawaban Ahli

Sebelum	Sesudah
[('variables', 'NNS'), ('well', 'RB'), ('declared', 'VBN'), ('anywhere', 'RB'), ('program', 'NN'), ('declared', 'VBN'), ('inside', 'IN'), ('function', 'NN'), ('local', 'JJ'), ('variables', 'NNS'), ('outside', 'IN'), ('functions', 'NNS'), ('global', 'JJ'), ('variables', 'NNS')]	[('variable', 'well', 'declare', 'anywhere', 'program', 'declare', 'inside', 'function', 'local', 'variable', 'outside', 'function', 'global', 'variable']

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

Setelah melakukan *pre-processing* pada dokument teks, akan didapatkan sekumpulan kata-kata yang kemudian akan dihitung bobotnya menggunakan algoritma TF-IDF. Langkah perhitungan TF-IDF dilakukan dengan cara menghitung banyaknya kemunculan *term* (TF) pada masing-masing dokumen jawaban peserta didik dan dokumen jawaban ahli. Lalu, nilai TF akan dinormalisasi dengan cara membagi nilainya dengan panjang dokumen. Selanjutnya ditentukan nilai *document frequency* (DF) dari masing-masing *term* yakni banyaknya dokumen dimana *term* tersebut muncul. Nilai ini akan digunakan dalam perhitungan IDF dengan mencari nilai log dari jumlah semua dokumen dibagi DF. Setelah mendapatkan nilai TF dan IDF masing-masing *term* di setiap dokumen, akan dilakukan perkalian kedua nilai untuk mendapatkan nilai TF-IDF. Contoh ilustrasi perhitungan untuk *lemmatization* dan *stemming* dapat dilihat pada tabel berikut.

Lemmatization

d1 = ['local', 'variable', 'well', 'declare', 'inside', 'method', 'function', 'use', 'begin', 'method', 'global', 'variable', 'declare', 'outside', 'function', 'method', 'generally', 'function', 'define']

q = ['variable', 'well', 'declare', 'anywhere', 'program', 'declare', 'inside', 'function', 'local', 'variable', 'outside', 'function', 'global', 'variable']

Tabel 3.15 TF-IDF *Lemmatization*

No	Term	TF		DF	D	IDF	TF*IDF	
		d1	q				d1	q
1	variable	0,1052631579	0,2142857143	2	2	0,1760912591	-0,01853592201	0,03773384123
2	well	0,05263157895	0,07142857143	2	2	0,1760912591	0,009267961003	0,01257794708
3	local	0,05263157895	0,07142857143	2	2	0,1760912591	0,009267961003	0,01257794708
4	declare	0,1052631579	0,1428571429	2	2	0,1760912591	-0,01853592201	0,02515589415
5	inside	0,05263157895	0,07142857143	2	2	0,1760912591	0,009267961003	0,01257794708

No	Term	TF		DF	D	IDF	TF*IDF	
		d1	q				d1	q
6	method	0,1578947368	0	1	2	0	0	0
7	function	0,1578947368	0,1428571429	2	2	0,1760912591	-0,02780388301	0,02515589415
8	use	0,05263157895	0	1	2	0	0	0
9	begin	0,05263157895	0	1	2	0	0	0
10	global	0,05263157895	0,07142857143	2	2	0,1760912591	0,009267961003	0,01257794708
11	outside	0,05263157895	0,07142857143	2	2	0,1760912591	0,009267961003	0,01257794708
12	generally	0,05263157895	0	1	2	0	0	0
13	define	0,05263157895	0	1	2	0	0	0
14	anywhere	0	0,07142857143	1	2	0	0	0
15	program	0	0,07142857143	1	2	0	0	0

Stemming

d1 = ['local', 'variabl', 'better', 'declar', 'insid', 'method', 'function', 'use', 'begin',
'method', 'global', 'variabl', 'declar', 'outsid', 'function', 'method', 'general',
'function', 'defin']

q = [variabl, 'well', 'declar', 'anywhere', 'program', 'declar', 'insid',
'functionlocal', 'variabl', 'outsid', 'function', 'global', 'variabl']

Tabel 3.16 TF-IDF *Stemming*

No	Term	TF		DF	D	IDF	TF*IDF	
		d1	q				d1	q
1	variabl	0,1052631579	0,2142857143	2	2	0,1760912591	-0,01853592201	0,03773384123
2	well	0	0,07142857143	2	2	0,1760912591	0	0,01257794708
3	better	0,05263157895	0	2	2	0,1760912591	0,009267961003	0
4	local	0,05263157895	0,07142857143	2	2	0,1760912591	0,009267961003	0,01257794708
5	declar	0,1052631579	0,1428571429	2	2	0,1760912591	-0,01853592201	0,02515589415
6	insid	0,05263157895	0,07142857143	2	2	0,1760912591	0,009267961003	0,01257794708

No	Term	TF		DF	D	IDF	TF*IDF	
		d1	q				d1	q
7	method	0,1578947368	0	1	2	0	0	0
8	function	0,1578947368	0,1428571429	2	2	0,1760912591	-0,02780388301	0,02515589415
9	use	0,05263157895	0	1	2	0	0	0
10	begin	0,05263157895	0	1	2	0	0	0
11	global	0,05263157895	0,07142857143	2	2	0,1760912591	0,009267961003	0,01257794708
12	outsid	0,05263157895	0,07142857143	2	2	0,1760912591	0,009267961003	0,01257794708
13	general	0,05263157895	0	1	2	0	0	0
14	defin	0,05263157895	0	1	2	0	0	0
15	anywher	0	0,07142857143	1	2	0	0	0
16	program	0	0,07142857143	1	2	0	0	0

Berdasarkan perhitungan di atas, didapatkan nilai TF-IDF untuk masing-masing term di d1 dan q pada proses *lemmatization* dan proses *stemming*.

3.2.3.8 Perhitungan Nilai Cosine Similarity

Tahap selanjutnya setelah mendapatkan bobot setiap kata adalah mencari nilai kemiripan dokumen menggunakan perhitungan *cosine similarity*. Pada perhitungan ini, variabel q atau *query* merupakan dokumen jawaban ahli (kunci) jawaban yang sebelumnya telah melalui tahap *pre-processing* dan pembobotan TF-IDF. Untuk mendapatkan nilai *cosine similarity* digunakan rumus pada persamaan 2.3.

$$Cos a = \frac{A.B}{|A||B|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (2.3)$$

Dimana dikalikan nilai A dan B sebagai nilai TF-IDF dua dokumen yang akan dihitung kemiripannya. Panjang A dan panjang B dihitung dengan dikuadratkan, menjumlahkan, lalu mengakarkan nilai TF-IDF *term* di masing-masing dokumen. Terakhir, hasil perkalian A dan B akan dibagi dengan nilai hasil perkalian panjang A dan B. Nilai akhir akan dikonversi menjadi nilai persentase. Contoh ilustrasi perhitungan dapat dilihat pada tabel berikut.

LemmatizationTabel 3.17 *Cosine Similarity Lemmatization*

No	Term	TF*IDF		Kuadrat		Weight
		d1	q	d1	q	d1*q
1	variable	-0,01853592201	-0,03773384123	0,0003435804046	0,001423842774	0,000699431538
2	well	-0,009267961003	-0,01257794708	0,00008589510115	0,0001582047526	0,000116571923
3	local	-0,009267961003	-0,01257794708	0,00008589510115	0,0001582047526	0,000116571923
4	declare	-0,01853592201	-0,02515589415	0,0003435804046	0,0006328190105	0,000466287692
5	inside	-0,009267961003	-0,01257794708	0,00008589510115	0,0001582047526	0,000116571923
6	method	0	0	0	0	0
7	function	-0,02780388301	-0,02515589415	0,0007730559104	0,0006328190105	0,000699431538
8	use	0	0	0	0	0
9	begin	0	0	0	0	0
10	global	-0,009267961003	-0,01257794708	0,00008589510115	0,0001582047526	0,000116571923
11	outside	-0,009267961003	-0,01257794708	0,00008589510115	0,0001582047526	0,000116571923
12	generally	0	0	0	0	0
13	define	0	0	0	0	0
14	anywhere	0	0	0	0	0
15	program	0	0	0	0	0
		Jumlah		0,001889692225	0,003480504558	0,002448010383

$$\begin{aligned}
 \text{Cos}(A, B) &= \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \\
 &= \frac{0,002448010383}{\sqrt{0,001889692225} \times \sqrt{0,003480504558}} \\
 &= \frac{0,002448010383}{0,04347059035 \times 0,05899580119} = 0,9545
 \end{aligned}$$

Stemming

Tabel 3.18 *Cosine Similarity Stemming*

No	Term	TF*IDF		Kuadrat		Weight
		d1	q	d1	q	d1*q
1	variabl	-0,01853592201	-0,03773384123	0,0003435804046	0,001423842774	0,000699431538
2	well	0	-0,01257794708	0	0,0001582047526	0
3	better	-0,009267961003	0	0,00008589510115	0	0
4	local	-0,009267961003	-0,01257794708	0,00008589510115	0,0001582047526	0,000116571923
5	declar	-0,01853592201	-0,02515589415	0,0003435804046	0,0006328190105	0,000466287692
6	insid	-0,009267961003	-0,01257794708	0,00008589510115	0,0001582047526	0,000116571923
7	method	0	0	0	0	0
8	function	-0,02780388301	-0,02515589415	0,0007730559104	0,0006328190105	0,000699431538
9	use	0	0	0	0	0
10	begin	0	0	0	0	0
11	global	-0,009267961003	-0,01257794708	0,00008589510115	0,0001582047526	0,000116571923
12	outsid	-0,009267961003	-0,01257794708	0,00008589510115	0,0001582047526	0,000116571923
13	general	0	0	0	0	0
14	defin	0	0	0	0	0
15	anywher	0	0	0	0	0
16	program	0	0	0	0	0
		Jumlah		0,001889692225	0,003480504558	0,00233143846

$$\begin{aligned}
 \text{Cos}(A, B) &= \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \\
 &= \frac{0,00233143846}{\sqrt{0,001889692225} \times \sqrt{0,003480504558}} \\
 &= \frac{0,00233143846}{0,04347059035 \times 0,05899580119} = 0,9091
 \end{aligned}$$

Berdasarkan perhitungan *cosine similarity* yang telah dilakukan, didapatkan nilai *cosine similarity* sebesar 0,9545 untuk model dengan penerapan *lemmatization* dan nilai sebesar 0,9091 untuk model dengan penerapan *stemming*.

3.2.3.9 Konversi Skor

Konversi skor diperlukan untuk membandingkan nilai model dan nilai perhitungan manual. Hal ini disebabkan oleh hasil dari perhitungan pada tahap *cosine similarity* memiliki rentang nilai 0-1. Berdasarkan penelitian Alfirna Rizqi Lahitani berjudul “*Automated Essay Scoring* menggunakan *Cosine Similarity* pada Penilaian Esai Multi Soal” pada tahun 2022 [3], konversi skor dari *cosine similarity* dapat dilakukan dengan cara mengalikan bobot kemiripan *cosine similarity* dengan bobot soal.

Nilai *Cosine similarity* dari model dengan penerapan *lemmatization* bernilai 0,9545 dan dengan penerapan *stemming* bernilai 0,9091. Bobot untuk soal bernilai 100 sehingga perhitungan konversi dapat dilakukan seperti berikut.

Tabel 3.19 Konversi Skor Jawaban Siswa dengan *Lemmatization*

Keterangan	Nilai
Jawaban siswa	0,9545
Bobot 100 untuk 1 soal	95,45
Skor Akhir	95,45

Tabel 3.20 Konversi Skor Jawaban Siswa dengan *Stemming*

Keterangan	Nilai
Jawaban siswa	0,9091
Bobot 100 untuk 1 soal	90,91
Skor Akhir	90,91

Berdasarkan konversi yang telah dilakukan, didapatkan nilai skor atau nilai model untuk penerapan *lemmatization* sebesar 95,45 dan untuk penerapan *stemming* sebesar 90,91.

3.2.4 Analisis Hasil

Setelah melakukan perancangan model, akan didapatkan hasil berupa nilai model untuk masing-masing penerapan proses *stemming* dan *lemmatization*. Selanjutnya, akan dilakukan analisis terhadap hasil yang telah didapatkan melalui evaluasi performa masing-masing metode dan perbandingan tingkat akurasi di antara keduanya.

3.2.4.1 Menghitung *Mean Absolute Error* (MAE)

Evaluasi performa model dilakukan dengan menghitung nilai *Mean Absolute Error* (MAE) pada masing-masing metode. Nilai ini didapatkan dengan cara menghitung selisih antara nilai hasil skoring oleh model dan nilai hasil penilaian dari evaluator manusia dibagi total jawaban yang dievaluasi seperti pada rumus 2.4. Setelah melakukan perhitungan evaluasi performa masing-masing model, maka akan didapatkan tingkat akurasi untuk model dengan penerapan proses *stemming* dan model dengan penerapan proses *lemmatization*.

Setelah mendapatkan nilai untuk masing-masing model, keduanya akan diuji menggunakan perhitungan *Mean Absolute Error* untuk mendapatkan tingkat akurasi. Perhitungan MAE didapatkan dengan menghitung selisih antara rata-rata nilai model dan rata-rata nilai perhitungan manual oleh evaluator manusia dibagi dengan jumlah seluruh dokumen jawaban peserta didik. Nilai perhitungan manual oleh evaluator manusia diasumsikan sebagai 100. Berikut merupakan perhitungan MAE untuk model dengan penerapan *stemming* dan model dengan penerapan *lemmatization*.

Lemmatization

$$MAE = \frac{\sum |n_{uji} - n_{model}|}{Jum} = \frac{\sum 100 - 95,45}{1} = 4,545454545$$

Stemming

$$MAE = \frac{\sum |n_{uji} - n_{model}|}{Jum} = \frac{\sum 100 - 90,91}{1} = 9,090909091$$

Nilai MAE dari model dengan penerapan *lemmatization* bernilai 4,545 dan nilai MAE dari model dengan penerapan *stemming* bernilai 9,091.

3.2.4.2 Perbandingan Tingkat Akurasi

Pada evaluasi performa, telah didapatkan tingkat akurasi untuk model dengan penerapan *stemming* dan model dengan penerapan *lemmatization*. Pada tahap ini, akan

dilakukan perbandingan dari tingkat akurasi keduanya untuk mengetahui tingkat akurasi dari masing-masing model.

Telah didapatkan nilai MAE untuk masing-masing model dengan penerapan *lemmatization* yaitu 4,545 dan *stemming* yaitu 9,091. Berdasarkan nilai tersebut, dapat disimpulkan bahwa model dengan penerapan *lemmatization* memiliki nilai MAE lebih rendah dibandingkan dengan model dengan penerapan *stemming*. Hal ini menunjukkan bahwa model dengan penerapan *lemmatization* memiliki performa yang lebih baik dibandingkan model dengan penerapan *stemming*.

3.3 Alat dan Bahan Tugas Akhir

Berikut merupakan alat dan bahan yang digunakan untuk membantu proses penelitian tugas akhir.

3.3.1 Alat

1. *Laptop* dengan spesifikasi sistem operasi Windows 10, *processor* Intel Core i3 6006U CPU @ 2 GHz, memori 12GB DDR4, grafis Intel(R) HD Graphics 520, SSD 125GB.
2. Python versi 3.9.7
3. Google Colab dengan modul NLTK, sklearn, pandas, numpy.

3.3.2 Bahan

1. Dataset pihak lain yang diperoleh dengan izin atau dalam lisensi yang diizinkan untuk digunakan secara langsung.
2. Dokumen panduan yang mengacu pada standar, hasil tugas akhir, atau artikel yang disitasi dan digunakan.

DAFTAR PUSTAKA

- [1] V. Kumala Sari, “KOLABORASI BAHASA INDONESIA DAN BAHASA INGGRIS SEBAGAI RAGAM BAHASA ILMU PENGETAHUAN DI ERA GLOBAL (Collaboration of Indonesian and English Languages as A Diversity of Knowledge Science In The Global Era).”
- [2] N. Chamidah, M. Mega Santoni, H. Nurramdhani Irmanda, R. Astriratma, F. Ilmu Komputer, and U. Pembangunan Nasional Veteran Jakarta, “Penilaian Esai Pendek Otomatis dengan Pencocokan Kata Kunci Frasa Nomina Short Essay Autoscoring using Noun Phrase-Keyword Matching,” 2021.
- [3] A. R. Lahitani, “Automated Essay Scoring menggunakan Cosine Similarity pada Penilaian Esai Multi Soal,” 2022. [Online]. Available: <http://ejurnal.ubharajaya.ac.id/index.php/JKI>
- [4] P. Penelitian dan Pengabdian Masyarakat Politeknik Transportasi Darat Bali, R. Ahmad, R. Rifai Oktavianus Sasue Teknologi Mekanika Otomotif, P. Transportasi Darat Bali, and J. Cempaka Putih Desa Samsam Kecamatan Kerambitan Kabupaten Tabanan, “SISTEM PENILAIAN ESAI OTOMATIS MENGGUNAKAN ALGORITMA STEMMING NAZIEF DAN ADRIANI,” *Jurnal Teknologi Transportasi dan Logistik*, vol. 1, no. 2, pp. 101–108, 2020.
- [5] N. Chamidah and M. Mega Santoni, “Pencocokan Berbasis Kata Kunci pada Penilaian Esai Pendek Otomatis Berbahasa Indonesia Keyword Matching-Based on Short Essay Autograding in Indonesian,” 2021.
- [6] S. Bahri and R. Wajhillah, “Optimalisasi Algoritma Rabin Karp menggunakan TF-IDF Dalam Pencocokan Text Pada Penilaian Ujian Essay Otomatis,” *Jl. Cemerlang*, vol. 4, no. 2, 2020, doi: 10.30743/infotekjar.v4i2.2294.
- [7] A. Riyani, M. Zidny Naf'an #2, and A. Burhanuddin, “Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen,” 2019.
- [8] R. I. Pratama, Munir, and R. Megasari, “Detector Similarity Answers Between Students on Essay Digital Exam System,” in *Proceedings of the 7th Mathematics, Science, and Computer Science Education International Seminar, MSCEIS 2019*, 2020. doi: 10.4108/eai.12-10-2019.2296348.

- [9] 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," in *Proceedings - CAMP 2021: 2021 5th International Conference on Information Retrieval and Knowledge Management: Digital Technology for IR 4.0 and Beyond*, Jun. 2021, pp. 70–74. doi: 10.1109/CAMP51653.2021.9498119.
- [10] R. Gupta and A. G. Jivani, "LemmaChase: A Lemmatizer," *International Journal on Emerging Technologies*, vol. 11, no. 2, pp. 817–824, 2020, [Online]. Available: www.researchtrend.net
- [11] I. Boban, A. Doko, and S. Gotovac, "Sentence retrieval using Stemming and Lemmatization with different length of the queries," *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 3, pp. 349–354, Jun. 2020, doi: 10.25046/aj050345.
- [12] I. Muftid, S. Lestanti, and ma Kholila, "APLIKASI PENILAIAN JAWABAN ESAI OTOMATIS MENGGUNAKAN METODE SYNONYM RECOGNITION DAN COSINE SIMILARITY BERBASIS WEB," 2021.
- [13] S. Burrows, I. Gurevych, and B. Stein, "The eras and trends of automatic short answer grading," *International Journal of Artificial Intelligence in Education*, vol. 25, no. 1. Springer New York LLC, pp. 60–117, Jan. 10, 2015. doi: 10.1007/s40593-014-0026-8.
- [14] M. N. Cholis, E. Yudaningtyas, and M. Aswin, "Pengaruh Penggunaan Synonym Recognition dan Spelling Correction pada Hasil Aplikasi Penilaian Esai dengan Metode Longest Common Subsequence dan Cosine Similarity," *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, vol. 3, no. 2, pp. 138–142, Mar. 2019, doi: 10.30743/infotekjar.v3i2.1061.
- [15] Gunawansyah, R. Rahayu, Nurwathi, B. Sugiarto, and Gunawan, "Automated essay scoring using natural language processing and text mining method," in *Proceeding of 14th International Conference on Telecommunication Systems, Services, and Applications, TSSA 2020*, Nov. 2020. doi: 10.1109/TSSA51342.2020.9310845.

- [16] M. A. Hussein, H. Hassan, and M. Nassef, "Automated language essay scoring systems: A literature review," *PeerJ Comput Sci*, vol. 2019, no. 8, 2019, doi: 10.7717/peerj-cs.208.
- [17] A. Wiratmo and C. Fatichah, "Indonesian Short Essay Scoring Using Transfer Learning Dependency Tree LSTM," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 2, pp. 278–285, Jan. 2020, doi: 10.22266/ijies2020.0430.27.
- [18] C. M. Ormerod, A. Malhotra, and A. Jafari, "Automated essay scoring using efficient transformer-based language models," 2021. [Online]. Available: <https://github.com/edx/ease>
- [19] T. Wahyuningsih, "Text Mining an Automatic Short Answer Grading (ASAG), Comparison of Three Methods of Cosine Similarity, Jaccard Similarity and Dice's Coefficient," *Journal of Applied Data Sciences*, vol. 2, no. 2, pp. 45–54, 2021.
- [20] "Study Design and Implementation of NLP Techniques for Automated Grading of answers: A Conceptual Model," 2021.
- [21] A. Tyagi, "A Review Study of Natural Language Processing Techniques for Text Mining." [Online]. Available: www.ijert.org
- [22] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, "Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations," *Organ Res Methods*, vol. 25, no. 1, pp. 114–146, Jan. 2022, doi: 10.1177/1094428120971683.
- [23] P. Gumaste, S. Joshi, S. Khadpekar, and S. Mali, "AUTOMATED QUESTION GENERATOR SYSTEM USING NLP LIBRARIES," *International Research Journal of Engineering and Technology*, 2020, [Online]. Available: www.irjet.net
- [24] M. J. Hazar, Z. H. Toman, and S. H. Toman, "Automated Scoring for Essay Questions in E-learning," in *Journal of Physics: Conference Series*, Oct. 2019, vol. 1294, no. 4. doi: 10.1088/1742-6596/1294/4/042014.
- [25] D. Khyani, S. B. S, N. N. M, and D. B. M, "An Interpretation of Lemmatization and Stemming in Natural Language Processing."

- [26] V. Chavan, S. R. Sirsat, and H. S. Mahalle, "Strength and Accuracy Analysis of Affix Removal Stemming Algorithms." [Online]. Available: <https://www.researchgate.net/publication/280933739>
- [27] M. Anjali and G. Jivani, "A Comparative Study of Stemming Algorithms." [Online]. Available: www.ijcta.com
- [28] H. Kiran Yadla and D. Prasada Rao, "Machine Learning Based Text Classifier Centered On TF-IDF Vectoriser", [Online]. Available: www.ijstr.org
- [29] R. G. Radu, I. M. Radulescu, C. O. Truica, E. S. Apostol, and M. Mocanu, "Clustering Documents using the Document to Vector Model for Dimensionality Reduction," in *2020 22nd IEEE International Conference on Automation, Quality and Testing, Robotics - THETA, AQTR 2020 - Proceedings*, May 2020. doi: 10.1109/AQTR49680.2020.9129967.
- [30] E. L. Amalia¹, A. J. Jumadi, I. A. Mashudi³, W. Wibowo⁴, and P. N. Malang, "ANALISIS METODE COSINE SIMILARITY PADA APLIKASI UJIAN ONLINE ESAI OTOMATIS (STUDI KASUS JTI POLINEMA) COSINE SIMILARITY METHOD ANALYSIS ON AUTOMATIC ESAI ONLINE TEST APPLICATION", doi: 10.25126/jtiik.202184356.
- [31] Moch. I. Anam, "sistem koreksi jawaban uraian singkat otomatis menggunakan metode cosine similarity dan qurey expansion," 2020.