TESIS

PERINGKASAN TEKS OTOMATIS PADA MODUL PEMBELAJARAN DI UNIVERSITAS TEKNOLOGI MATARAM MENGGUNAKAN METODE LATENT SEMANTIC ANALYSIS (LSA)

Disusun oleh:

Nama : **ST Tuhpatussania**
NIM : **21.55.1026**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2023

TESIS

PERINGKASAN TEKS OTOMATIS PADA MODUL PEMBELAJARAN DI UNIVERSITAS TEKNOLOGI MATARAM MENGGUNAKAN METODE LATENT SEMANTIC ANALYSIS (LSA)

AUTOMATIC TEXT SUMMARIZATION IN LEARNING MODULES AT THE UNIVERSITY OF TECHNOLOGY MATARAM USING LATENT SEMANTIC ANALYSIS (LSA) METHOD

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama : **ST Tuhpatussania**
NIM : **21.55.1026**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2023

HALAMAN PENGESAHAN

PERINGKASAN TEKS OTOMATIS PADA MODUL PEMBELAJARAN DI UNIVERSITAS TEKNOLOGI MATARAM MENGGUNAKAN METODE LATENT SEMANTIC ANALYSIS (LSA)

AUTOMATIC TEXT SUMMARIZATION IN LEARNING MODULES AT THE UNIVERSITY OF TECHNOLOGY MATARAM USING LATENT SEMANTIC ANALYSIS (LSA) METHOD

Dipersiapkan dan Disusun oleh

ST Tuhpatussania

21.55.1026

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Jum'at, 03 Maret 2023

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 03 Maret 2023
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

PERINGKASAN TEKS OTOMATIS PADA MODUL PEMBELAJARAN DI UNIVERSITAS TEKNOLOGI MATARAM MENGGUNAKAN METODE LATENT SEMANTIC ANALYSIS (LSA)

AUTOMATIC TEXT SUMMARIZATION IN LEARNING MODULES AT THE UNIVERSITY OF TECHNOLOGY MATARAM USING LATENT SEMANTIC ANALYSIS (LSA) METHOD

Dipersiapkan dan Disusun oleh

ST Tuhpatussania

21.55.1026

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Jum'at, 03 Maret 2023

Pembimbing Utama	**Anggota Tim Penguji**
Prof. Dr. Ema Utami, S.Si., M.Kom	**Dr. Andi Sunyoto, M.Kom.**
NIK. 190302037	**NIK. 190302052**
	Hanafi, S.Kom., M.Eng., Ph.D.
Pembimbing Pendamping	**NIK. 190302024**
Anggit Dwi Hartanto, M.Kom.	**Prof. Dr. Ema Utami, S.Si., M.Kom**
NIK. 190302163	**NIK. 190302037**

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 03 Maret 2023
Direktur Program Pascasarjana

Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : ST Tuhpatussania
NIM : 21.55.1026
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Peringkasan Teks Otomatis Pada Modul Pembelajaran di Universitas Teknologi Mataram Menggunakan Metode Latent Semantic Analysis (LSA)

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom
Dosen Pembimbing Pendamping : Anggit Dwi Hartanto, M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 03 Maret 2023
Yang Menyatakan,

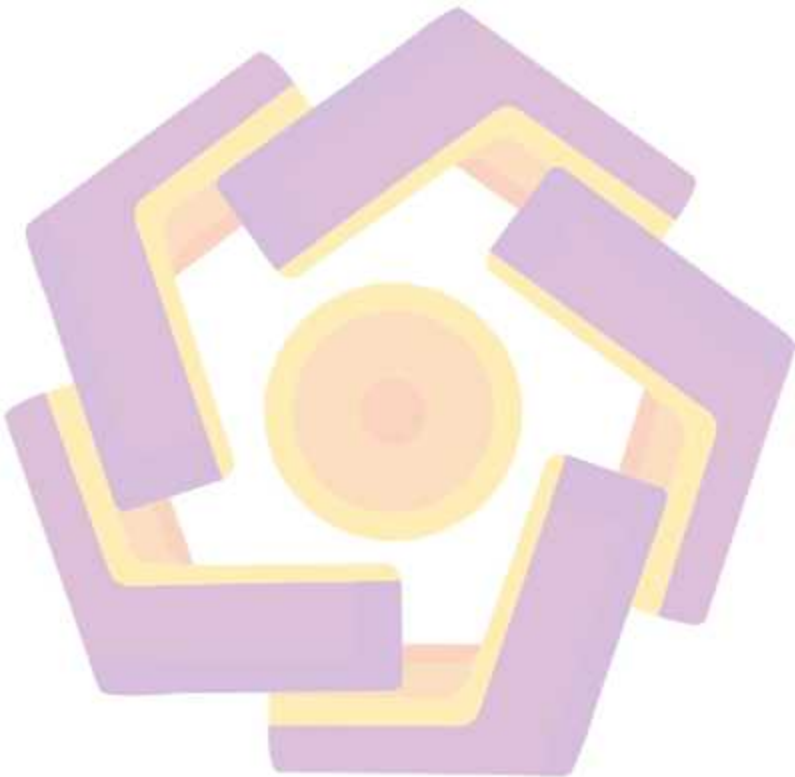
ST Tuhpatussania

HALAMAN PERSEMBAHAN

Tesis ini saya persembahkan untuk Bapak Ir. H. Lalu Darmawan Bakti, M.Sc., M.Kom dan ibu Dwinita Arwidiyarti, M.Kom karena beliau berdua lah yang selalu memberikan saya *support system* baik dari segi finansial maupun dari segi semangat yang berupa omelan. Semoga Tesis ini bermanfaat untuk pengembangan penelitian-penelitian selanjutnya.

HALAMAN MOTTO

"BEBAS, MERDEKA !!!"



KATA PENGANTAR

Alhamdulillah, segala puji dan syukur penulis panjatkan kehadirat Allah SWT karena atas segala karunia dan ridho-Nya, sehingga tesis yang berjudul "Peringkasan Teks Otomatis pada Modul Pembelajaran di Universitas Teknologi Mataram Menggunakan Metode Latent Semantic Analysis (LSA)" dapat diselesaikan dengan tepat waktu.

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar Magister Komputer pada program studi Magister Teknik Informatika Universitas Amikom Yogyakarta.

Penyelesaian tesis yang sangat berharga ini tidak lepas dari bantuan dan dukungan dari berbagai pihak. Pada kesempatan ini, penulis mengucapkan rasa syukur dan terima kasih kepada:

1. Rektor dan Civitas Akademika UTM yang telah memberikan saya ruang dan kesempatan untuk menempuh hingga menyelesaikan studi Magister ini.
2. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom selaku pembimbing utama yang telah membimbing, membantu, dan memotivasi dalam penulisan tesis ini sehingga dapat terselesaikan dengan baik.
3. Bapak Anggit Dwi Hartanto, M.Kom. selaku pembimbing pendamping yang telah membimbing, membantu, dan memotivasi dalam penulisan tesis ini sehingga dapat terselesaikan dengan baik.
4. Dosen Penguji yang telah memberikan saran yang baik demi kemajuan tesis ini.
5. Direktur Program Pascasarjana, jajarannya, staf dan rekan-rekan Magister Teknik Informatika Universitas Amikom Yogyakarta.

Yogyakarta, 03 Maret 2023

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

Peringkasan teks otomatis berbahasa Indonesia umumnya lebih rumit dibandingkan bahasa inggris karena untuk melakukan peringkasan teks otomatis dibutuhkan bobot dari setiap kata yang muncul dalam sebuah dokumen untuk dicari hubungan kontekstual antarkata dalam sebuah kalimat sehingga kata yang muncul dalam sebuah dokumen perlu di ubah ke asal katanya (*root word*) agar tidak terjadi redudansi. Tujuan melakukan peringkasan teks otomatis adalah untuk membantu seseorang membaca suatu teks secara ringkas dengan menghasilkan ringkasan secara otomatis dari suatu teks tanpa adanya proses penyuntingan manusia terhadap ringkasan tersebut. Pada penelitian ini penulis meringkas modul pembelejaran teori pada Universitas Teknologi Mataram sejumlah 25 modul pembelajaran berbahasa Indonesia menggunakan metode LSA (*Latent Semantic Analysis*) untuk tahapan peringkasan teks dan metode TF-IDF (*Term frequency-Inverse document frequency)* pada tahapan pembobotan kata serta membandingkan metode *stemming* Nazief-Adriani dan *stemming* Porter dalam menentukan kata asal (*root word*).

Modul pembelajaran yang digunakan hanya modul pembelajaran untuk matakuliah teori dengan tahapan peringkasan yaitu *Preprocessing*, pembobotan kata, peringkasan teks otomatis lalu evaluasi hasil menggunakan metode *ROUGE-1, ROUGE-2* dan *ROUGE-L* dan dibandingkan dengan hasil peringkasan oleh tiga pakar yaitu dua pakar dibidang ilmu sesuai dengan modul pembelajaran yang diringkas dan satu pakar dibidang Bahasa.

Hasil perbandingan metode stemming Nazief-Adriani dan stemming Porter selisih 13 detik lebih lama menggunakan stemming Nazief-Adriani dalam waktu pemrosesan data sedangkan akurasi hasil stemming lebih tinggi menggunakan metode Nazief-Adriani. Dalam peringkasan teks otomatis menggunakan LSA memiliki tingkat akurasi rata-rata 83,49% dengan Akurasi yang di hasilkan dapat lebih tinggi jika dibandingkan dengan penelitian terdahulu yaitu diatas 70%.

Kata kunci: peringkasan teks otomatis, LSA, Nazief-adriani, porter, NLP.

ABSTRACT

Automatic text summarization in Indonesian is generally more complicated than English because to do automatic text summarization it takes the weight of each word that appears in a document to look for contextual relationships between words in a sentence so that words that appear in a document need to be changed to the origin of the word (root). word) to avoid redundancies. The purpose of performing automatic text summarization is to help one read a text in a concise manner by automatically generating a summary of a text without any human editing of the summary. In this study the authors summarized the theory learning modules at the Mataram University of Technology with a total of 25 Indonesian language learning modules using the LSA (Latent Semantic Analysis) method for the text summary stage and the TF-IDF (Term frequency-Inverse document frequency) method at the word weighting stage and comparing methods Nazief-Adriani stemming and Porter stemming in determining root words.

The learning modules used are only learning modules for theoretical courses with summary stages, namely Preprocessing, word weighting, automatic text summarization and then evaluation of results using the ROUGE-1, ROUGE-2 and ROUGE-L methods and compared with the summarization results by three experts, namely two experts in the field knowledge in accordance with the summarized learning modules and one expert in the field of language.

The results of the comparison of the Nazief-Adriani stemming method and Porter's stemming method differ by 13 seconds longer using Nazief-Adriani stemming in data processing time while the accuracy of the stemming results is higher using the Nazief-Adriani method. In summarizing automatic text using LSA it has an average accuracy rate of 83.49% with the resulting accuracy being higher when compared to previous research, which is above 70%.

Keywords: automatic text summarization, LSA, Nazief-adriani, porter, NLP.

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Perkembangan kemajuan teknologi yang pesat membuat manusia menjadi lebih mudah dalam menemukan informasi-informasi yang dibutuhkan. Baik itu informasi yang didapatkan dari membaca di internet maupun dari buku, modul, artikel, majalah, koran, dan lain-lain. Semakin banyak dan mudahnya informasi yang diperoleh tidak sejalan dengan minat baca masyarakat yang pada umumnya senang mendapatkan informasi secara instan padahal kegiatan yang wajib dilakukan guna memperkaya pengetahuan dan informasi ialah dengan cara membaca dari banyak referensi. Salah satu solusi untuk mendapatkan inti informasi dari keseluruhan artikel maupun modul secara cepat dan menghemat waktu adalah dengan membaca bagian ringkasan dari dokumen untuk itu peringkasan teks otomatis sangat di perlukan.

Peringkasan teks otomatis merupakan teknologi yang membantu seseorang untuk membaca suatu teks secara ringkas dengan menghasilkan ringkasan secara otomatis dari suatu teks tanpa adanya proses penyuntingan manusia terhadap ringkasan tersebut (Firman et al., 2022). Sehingga dengan menggunakan peringkasan teks otomatis akan mendapatkan ide pokok maupun informasi penting yang dibutuhkan dari dokumen dengan jelas, tepat, dan ringkas, tanpa merubah maknanya. Peringkasan teks sendiri pada umumnya terdiri dari dua teknik yaitu, peringkasan teks ekstraksi dan abstraksi. Peringkasan ekstraktif memilih beberapa

kalimat dari dokumen asli untuk merepresentasikan dokumen secara keseluruhan tanpa mengubah struktur kalimat-kalimat tersebut(Halimah et al., 2022). Sedangkan peringkasan abstraktif menyusun ulang kalimat-kalimat menjadi ringkasan berdasarkan kata-kata inti yang terdapat pada dokumen asli. Peringkasan secara abstraktif lebih susah dilakukan daripada peringkasan secara ekstraktif (Husniah et al., 2022)

Peringkasan teks otomatis berbahasa indonesia umumnya lebih rumit dibandingkan berbahasa inggris karena untuk melakukan peringkasan teks otomatis dibutuhkan bobot dari setiap kata yang muncul dalam sebuah dokumen untuk dicari hubungan kontekstual antarkata dalam sebuah kalimat sehingga kata yang muncul dalam sebuah dokumen perlu di ubah ke asal katanya (*root word*) agar tidak terjadi redudansi (Satwika & Alam, 2020). Bahasa indonesia adalah bahasa yang kaya akan imbuhan, sehingga untuk merubah kata dalam Bahasa Indonesia ke kata dasar umumnya dilakukan dengan menghapus imbuhan katanya, ada kurang lebih 35 imbuhan yang disebutkan dalam Kamus Besar Bahasa Indonesia (Simanjuntak, 2022). Imbuhan-imbuhan ini dapat berupa prefix (awalan), surfix (akhiran), konfix maupun infix (sisipan) yang diserap dari bahasa Jawa. Pemakaian imbuhan dalam kata Bahasa Indonesia dapat digunakan pada semua kata, dan imbuhan-imbuhan tersebut dapat dikombinasikan dengan satu dan lainnya secara bebas, seperti afiksasi, reduplikasi, dan lainnya.

Penelitian terdahulu banyak yang membahas peringkasan teks otomatis (*Automatic Text* Summarization) dengan menggunakan berbagai metode, Beberapa diantaranya adalah penelitian oleh (Syahfitri et al., 2022), melakukan

peringkasan teks otomatis menggunakan algoritma *Maximum Marginal Relevance* dengan melakukan pembobotan untuk setiap kalimat yang ada pada dokumen. Hasil ringkasan dari sistem peringkasan teks otomatis pada penelitian ini adalah kalimat inti yang mirip dengan *query* dan berdasarkan urutan bobot sehingga tampilan akhir dari hasil peringkasan ini tidak memiliki urutan sistematik yang baik.

Yunita Maulidia Sari juga melakukan penelitian terkait peringkasan modul pembelajaran berbahasa Indonesia menggunakan metode Cross *Latent Semantic Analysis* (CSLA). pengujian akurasi pada peringkasan modul pembelajaran dilakukan dengan cara membandingkan hasil ringkasan manual oleh manusia dan hasil ringkasan sistem. Yang mana pengujian ini menghasilkan rata-rata nilai *f-measure, precision,* dan *recall* tertinggi pada *compression rate* 20% dengan nilai berturut-turut 0.3853, 0.432, dan 0.3715 (Sari & Nenden, 2021).

Penelitian lain yang masih terkait dalam penelitian ini dilakukan oleh (Nabilah, 2022) dengan judul penelitian "Peringkasan Teks Bahasa Indonesia Pada Cerpen Menggunakan Metode *Latent Semantic Analysis* (LSA)". pada penelitian tersebut menggunakan data uji sebanyak 30 teks cerpen berbahasa Indonesia dan melakukan peringkasan dengan *compression rate* 30% dan mendapatkan hasil evaluasi *recall* 68,4%, *f-measure* 71,43% dan *precision* 74,93%, meski hasil perhitungan evaluasi belum cukup tinggi namun peringkasan sudah dapat dikatakan memberikan hasil ringkasan yang menyerupai hasil ringkasan manual dengan cukup baik dalam mendeskripsikan isi cerpen secara keseluruhan.

Algoritma LSA juga sering digunakan untuk Teknik peringkasan teks berbahasa Indonesia, salah satunya penilitian yang dilakukan oleh (Rozi et al.,

2021) yang menggunakan metode SVD (*Singular Value Decomposition*) pada algoritma LSA untuk menghitung kesamaan kata antar kalimat dan menghitung panjang length pada matriks yang diperoleh dari perhitungan metode SVD. Hasil pengujian terhadap 10 dokumen hukum berbahasa indonesia yang diringkas oleh pakar, diperoleh hasil *precision, recall, f-measure dan accuracy* secara berurutan pada peringkasan teks otomatis dengan metode *Latent Semantic Analysis* untuk *compression rate* 75% yaitu 53%, 27%, 35% dan 71% lalu untuk *compression rate* 50% yaitu 54%, 56%, 55% dan 75%, serta untuk *compression rate* 25% yaitu 51%, 79%, 61% dan 75%.

Penelitian lain yang masih terkait dalam penelitian ini juga dilakukan oleh (Alvida et al, 2020) yang melakukan penelitian terkait metode stemming Porter dengan judul penelitian dalam Bahasa inggris "*Identification of topics in News Articles Using Algorithm of Porter Stemmer Enhancement and Likelihood Classifier*". Penelitian tersebut melakukan klasifikasi berita berdasarkan kategori dan identifikasi topik berita dengan menerapkan algoritma Porter Stemmer Enhancement dalam proses stemming dan Metode Likelihood untuk klasifikasi Berita. Berdasarkan hasil pengujian dari 900 data latih dan 90 data uji, diperoleh hasil yang cukup baik akurasi tinggi, yaitu 95,56% untuk klasifikasi kategori dan 97,78% untuk topik identifikasi.

Penelitian terdahulu terkait metode stemming Bahasa Indonesia juga pernah dilakukan, dengan judul penelitian dalam Bahasa inggris "*Comparison of Stemming Test Results of Tala Algorithms with Nazief Adriani in Abstract Documents and National News*" (Pamungkas et al, 2023) pada penelitian tersebut melakukan

perbandingan metode stemming Nazief-Adriani dan metode stemming Tala. Kedua metode stemming tersebut memiliki aturan pencarian kata dasar yang berbeda. Hasil pengujian yang telah dilakukan dapat disimpulkan bahwa algoritma stemming Tala memiliki tingkat akurasi yang lebih rendah dari Nazief-Adriani. Algoritma Tala hanya memiliki akurasi rata-rata sebesar 65,29%, sedangkan Nazief Adriani memiliki akurasi sebesar 78,47%. Mengenai kecepatan, algoritma Tala memiliki kecepatan yang lebih baik dari Nazief Adriani sebesar 32,19 detik dan Nazief & Adriani sebesar 65,2 detik.

Berlandaskan kebutuhan terhadap perlunya peringkasan teks otomatis memotivasi peneliti untuk melakukan peringkasan teks otomatis dengan dataset modul pembelajaran berbahasa Indonesia di Universitas Teknologi Mataram, dan berdasarkan penelitian terdahulu terkait peringkasan teks otomatis modul pembelajaran dengan metode CLSA (*Cross Latent Semantic Analysis*) yang menghasilkan nilai akurasi rendah maka pada penelitian ini peringkasan teks otomatis modul pembelajaran berbahasa Indonesia akan menggunakan metode *Latent Semantic Analysis* (LSA). Dan untuk membantu kinerja metode LSA agar hasil peringkasan teks lebih baik dari penelitian terdahulu maka dalam peringkasan teks otomatis modul pembelajaran bahasa indonesia ini yang rata-rata dokumennya terdiri dari teks panjang berisikan pikiran, pendapat dan materi pembelajaran selama satu semester atau 6 bulan sehingga susunan katanya tentu banyak memiliki makna sinonim dan perlu melalui tahap stemming atau percarian kata dasar, untuk itu peneliti akan menggunakan dua algoritma stemming dan membandingkannya

yaitu algoritma Nazief-Adriani yang berbasis kamus Bahasa Indonesia dan algoritma Porter yang berbasis penghapusan imbuhan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang sudah di uraikan diatas, permasalahan yang dapat diangkat adalah sebagai berikut:

a. Algorima *stemming* manakah yang mempengaruhi hasil akurasi lebih tinggi pada peringkasan teks otomatis modul pembelajaran berbahasa Indonesia antara algoritma *stemming* Porter dan algoritma *stemming* Nazief-Adriani?
b. Berapa tingkat akurasi yang dihasilkan dengan menggunakan metode *Latent Semantic Analysis* dalam peringkasan teks otomatis modul pembelajaran berbahasa Indonesia ?

1.3. Batasan Masalah

Untuk membatasi ruang lingkup yang terlalu luas atau melebar serta agar penelitian ini bisa lebih fokus, maka perlu dilakukan pembatasan masalah. Berikut adalah batasan masalah dalam penelitian ini:

a. Sample data sebanyak 25 file dari Universitas Teknologi Mataram berupa modul pembelajaran berbahasa indonesia yang berformat pdf dan docx serta file tidak dikunci.
b. Pada tahap *stemming* menggunakan dua metode yaitu metode Porter dan metode Nazief-Adriani serta pada tahap pembobotan kata menggunakan metode TF-IDF dan tahap peringkasan teks menggunakan metode *Latent Semantic Analysis* (LSA).

c. Pengujian pada penelitian ini menggunakan metode ROUGE untuk menghitung precision, recall dan f-measure terhadap peringkasan teks dari hasil sistem dan dari pakar.

1.4. Tujuan Penelitian

a. Mengetahui berbandingan algoritma Porter dan algoritma Nazief-Adriani pada tahapan *stemming* yang memiliki performa lebih baik dalam penerapannya untuk peringkasan teks otomatis modul pembelajaran menggunakan algoritma *Latent Semantic Analysis*.

b. Menghasilkan Metode peringkasan teks otomatis untuk data berupa modul pembelajaran dengan tingkat akurasi yang tinggi menggunakan algoritma *Latent Semantic Analysis*.

1.5. Manfaat Penelitian

a. Menambah wawasan terkait aplikasi peringkasan teks otomatis modul pembelajaran menggunakan metode LSA.

b. Dapat memudahkan peringkasan teks secara otomatis dengan tingkat akurasi yang tinggi.

c. Memudahkan mahasiswa atau mahasiswi dalam memahami inti informasi pada modul pembelajaran serta memudahkan dosen dalam proses perkuliahan jika mahasiswa atau mahasiswi sudah memahami garis beras materi perkuliahan.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Beberapa penelitian yang pernah dilakukan yang berkaitan dengan tema yang diambil berdasarkan metode yang digunakan dan objek yang digunakan.

Ayu Syahfitri R, Ade Kurniawan dan Mohd Ihsan Humaidy telah melakukan penelitian tentang Penerapan Algoritma Maximum Marginal Relevance Dalam Peringkasan Teks Secara Otomatis (Syahfitri et al., 2022) dimana proses secara umum yang dilakukan dalam pembuatan ringkasan otomatis pada penelitian ini, yaitu *text preprosessing* meliputi pemecahan kalimat, *case folding, filtering, tokenizing* kata dan *stemming*. Serta proses pembobotan dengan TF-IDF dan peringkasan teks menggunakan algoritma MMR. Algoritma *maximum marginal relevance* (MMR) digunakan untuk merangking kalimat-kalimat sebagai tanggapan terhadap *query* yang diberikan oleh *user*. Perhitungan MMR dilakukan dengan iterasi dengan mengkombinasikan dua matrik *cosine similarity* yaitu relevansi antara *query* terhadap keseluruhan kalimat dan *similarity* antara kalimat dengan kalimat. Hasil ringkasan dari sistem peringkasan teks otomatis pada penelitian ini adalah kalimat inti yang mirip dengan query dan berdasarkan urutan bobot, jadi untuk pengembangan penelitian berikutnya diharapkan hasil ringkasan memiliki urutan berdasarkan sistematika yang baik agar lebih mudah dipahami oleh masyarakat luas.

Yunita Maulidia Sari dan Nenden Siti Fatonah telah melakukan penelitian Peringkasan Teks Otomatis pada Modul Pembelajaran Berbahasa Indonesia Menggunakan Metode *Cross Latent Semantic Analysis* (CLSA). Jumlah data yang digunakan pada penelitian ini sebanyak 10 file modul pembelajaran yang berasal dari modul para dosen Universitas Mercu Buana, dengan format .docx sebanyak 5 file dan format .pdf sebanyak 5 file. Penelitian ini menerapkan metode Term Frequency-Inverse Document Frequency (TF-IDF) untuk pembobotan kata dan metode *Cross Latent Semantic Analysis* (CLSA) untuk peringkasan teks. Pengujian akurasi pada peringkasan modul pembelajaran dilakukan dengan cara membandingkan hasil ringkasan manual oleh manusia dan hasil ringkasan sistem. Yang mana pengujian ini menghasilkan rata-rata nilai *f-measure, precision, dan recall* tertinggi pada *compression rate* 20% dengan nilai berturut-turut 0.3853, 0.432, dan 0.3715. Hasil pengujian tersebut mendapatkan nilai akurasi yang cukup rendah sehingga metode tersebut tidak cocok digunakan untuk data uji berupa file dokumen.

Malak Diana D, Bagas Satya D. N dan Faisal Rahutomo melakukan penelitian tentang Penerapan Algoritma *Score-Based* Pada Peringkasan Teks Cerpen Otomatis (Diana D et al., 2020), Algoritma yang digunakan yaitu algoritma score-based. Algoritma score-based tersebut digunakan untuk mencari inti dari teks dengan memberikan nilai untuk setiap kalimat dari perhitungan fitur yang telah ditentukan. Pengujian akurasi peringkasan teks cerpen menggunakan algoritma score-based dilakukan dengan membandingkan ringkasan sistem dengan ringkasan manual dari pakar dengan hasil rata-rata nilai *precision* sebesar 52, *recall* sebesar

44,88889, dan *f-measure* sebesar 46,65009. Pengujian kepuasan pembaca dilakukan menggunakan kuisioner dan diperoleh hasil sebesar 71,3%. Pada penelitian ini tidak melakukan pembobotan pada masing-masing fitur sehingga tidak diketahui fitur mana yang lebih penting dalam peringkasan teks otomatis cerpen tersebut.

Aa Zezen, Zainal Abidin dan Enung Nurjanah telah melakukan penelitian terkait Sistem Peringkas Teks Otomatis Multi Dokumen Kliping Artikel Berita Gempa Menggunakan Metode TF-IDF (Zezen et al., 2020), pada penelitian ini membuat *prototype* system peringkas teks multi dokumen menggunakan metode *Term Frequency Inverse Document Frequency* (TF-IDF) yaitu memecah isi dokumen menjadi kalimat, membuang karakter, memecah kalimat menjadi kata, memberi nilai bobot pada kata, menjumlahkan nilai bobot, menghitung nilai idf dan TF-IDF sehingga di dapat nilai bobot kata dari setiap kalimat, diperoleh bobot kalimat dimana bobot yang tertinggi atau beberpa kalimat dengan rangking tertinggi dijadikan ringkasan dari masing-masing dokumen. Ringkasan dari setiap dokumen digabung dan diringkaskan lagi, sehingga menjadi ringkasan ketiga sebagai gabungan dua dokumen. Digunakan tools berbasis web, dengan bahasa pemogramman PHP dan DBMS MySQL. Aplikasi ini dapat mengimplementasikan peringkas teks otomatis multi dokumen kliping artikel berita di internet metode TF-IDF. Sistem ini dapat membantu mengetahui isi penting dari kliping artikel berita yang banyak di internet. Memiliki akurasi hasil uji responden 54.45% dan uji kemiripan dokumen sebesar 78.023%.

Algoritma LSA juga sering digunakan untuk Teknik peringkasan teks berbahasa Indonesia, salah satunya penilitian yang dilakukan oleh (Rozi et al., 2021) yang menggunakan metode SVD (*Singular Value Decomposition*) pada algoritma LSA untuk menghitung kesamaan kata antar kalimat dan menghitung panjang length pada matriks yang diperoleh dari perhitungan metode SVD. Hasil pengujian terhadap 10 dokumen hukum berbahasa indonesia yang diringkas oleh pakar, diperoleh hasil *precision, recall, f-measure dan accuracy* secara berurutan pada peringkasan teks otomatis dengan metode Latent Semantic Analysis untuk compression rate 75% yaitu 53%, 27%, 35% dan 71% lalu untuk compression rate 50% yaitu 54%, 56%, 55% dan 75%, serta untuk compression rate 25% yaitu 51%, 79%, 61% dan 75%.

Penelitian lain yang menggunakan algoritma LSA pada peringkasan teks juga dilakukan oleh (Nabilah , 2022) yang melakukan peringkasan teks pada cerpen berbahasa Indonesia. Pada penelitian ini, peringkasan teks dilakukan dengan menggunakan algoritma Latent Semantic Analysis (LSA). Pengujian dilakukan dengan menggunakan data uji sebanyak 30 teks cerpen dan Compression Rate kalimat rangkuman sebanyak 30% dari jumlah kalimat teks cerpen. Pada penelitian ini terdapat tiga tahapan yaitu pertama *text preprocessing*, selanjutnya menghitung bobot TF-IDF, dan terakhir menentukan kalimat yang akan dirangkum dengan menghitung *Latent Semantic Analysis* (LSA). Tingkat hasil rangkuman teks diukur dengan perhitungan recall, precision, dan f-measure. Penelitian ini menghasilkan nilai teks dalam cerpen memiliki rata-rata *precison* 74,93%, *recall* 68,4%, dan *f-measure* 71,43%. Berdasarkan akurasi tersebut dapat dikatakan memberikan hasil

ringkasan yang menyerupai hasil ringkasan manual dengan cukup baik dalam mendeskripsikan isi cerpen secara keseluruhan.

Proses peringkasan teks otomatis berbahasa Indonesia pada umumnya melalui tahap preprocessing dimana pada tahapan tersebut terdapat proses stemming untuk membantu proses pembobotan kata agar tidak tidak terjadi duplikasi kata dalam makna yang sama. Oleh karena itu berikut ini beberapa penelitian terdahulu terkait metode stemming untuk Bahasa Indonesia yaitu oleh (Alvida et al, 2020) Penelitian tersebut menerapkan algoritma Porter Stemmer dalam proses stemming dan Metode Likelihood untuk klasifikasi berita berdasarkan kategori dan identifikasi topik karena setiap berita pasti memiliki karakteristik informasi yang berbeda sehingga diperlukan suatu algoritma khusus yang mampu menangani penemuan topik dan klasifikasi menggunakan data pelatihan pada banyak artikel berita Indonesia. Hasil yang optimal bila menggunakan jumlah kata kunci sebanyak 25, sedangkan untuk identifikasi topik didapatkan maksimal result dengan jumlah kata kunci sebanyak 20. Nilai akurasi untuk klasifikasi kategori diperoleh sebesar 95,56%, sedangkan untuk topik identifikasi adalah 97,78%.

Selain dari metode stemming Porter yang dapat digunakan pada peringkasan teks berbahasa Indonesia, juga terdapat metode stemming Nazief-Adriani seperti pada penelitian oleh (Natalinda et al, 2023) dengan judul dalam Bahasa inggris "*Comparison of Stemming Test Results of Tala Algorithms with Nazief Adriani in Abstract Documents and National News*". Pada penelitian tersebut, isi informasi yang diperoleh dari dokumen-dokumen kemudian disortir agar lebih mudah dipahami maknanya. Proses sortasi ini dikenal sebagai stemming. Stemming adalah

proses yang banyak diterapkan dalam pencarian kata dasar. Memisahkan kata-kata yang tidak berarti dapat membuat informasi menjadi lebih jelas. Perlu diperhatikan algoritma stemming yang tepat sesuai dengan bahasa yang digunakan. Banyak algoritma stemming yang dapat digunakan untuk melakukan proses pencarian kata dasar ini. Beberapa di antaranya adalah algoritma Tala dan Nazief Adriani. Kedua algoritma tersebut memiliki perbedaan dalam proses kerjanya. Algoritma Tala mengadopsi algoritma Porter berbasis aturan, sedangkan algoritma Nazief & Adriani bekerja berdasarkan kamus. Kedua algoritma tersebut memiliki keunggulan masing-masing dalam hal akurasi dan kecepatan. Oleh karena itu, pada penelitian ini akan dilakukan analisis dengan membandingkan kinerja kedua algoritma tersebut dalam proses text-stemming berbahasa Indonesia. Proses uji coba menggunakan beberapa sumber data yang berbeda untuk mengukur kecepatan dan akurasi dari masing-masing algoritma. Sumber data yang digunakan dalam penelitian ini antara lain abstrak laporan skripsi atau tugas akhir mahasiswa sebanyak 30 mahasiswa dan informasi dari berita online sebanyak 200. Dari hasil pengujian yang telah dilakukan dapat disimpulkan bahwa algoritma stemming Tala memiliki tingkat akurasi yang lebih rendah dari Nazief Adriani. Algoritma Tala hanya memiliki akurasi rata-rata sebesar 65,29%, sedangkan Nazief Adriani memiliki akurasi sebesar 78,47%. Mengenai kecepatan, algoritma Tala memiliki kecepatan yang lebih baik dari Nazief Adriani sebesar 32,19 detik dan Nazief & Adriani sebesar 65,2 detik.

Perbedaan penelitian ini jika di bandingkan dengan penelitian terdahulu adalah jumlah kata atau kalimat yang lebih banyak terkandung dalam modul

pembelajaran pada dataset yang akan digunakan serta penerapan algoritma Porter dan algoritma Nazief-Adriani pada tahap *stemming* yang belum pernah dibandingkan sebelumnya untuk peringkasan teks otomatis modul pembelajaran menggunakan *Latent Semantic Analysis*.

2.2. Keaslian Penelitian

Tabel 2. 1. Matriks *Literatur Review* dan Posisi Penelitian

Peringkasan Teks Otomatis Pada Modul Pembelajaran Menggunakan Metode *Latent Semantic Analysis* (LSA)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Penerapan Algoritma Maximum Marginal Relevance Dalam Peringkasan Teks Secara Otomatis	Ade Kurniawan, Mohd.Irsan Humaidy, BULLETINDS, 2022.	Melakukan peringkasan teks dengan menggunakan algoritma MMR dan pembobotan dengan TF-IDF.	Aplikasi peringkasan teks otomatis menggunakan metode algoritma maximum marginal relevance dapat mengahasilkan ringkasan secara otomatis tanpa menghilangkan makna dalam teks. Dapat menerapkan Algoritma maximum marginal relevance dalam melakukan peringkasan menggunakan aplikasi peringkas teks secara otomatis. Hasil ringkasan dari sistem peringkasan teks otomatis pada penelitian ini adalah kalimat inti yang mirip dengan *query* dan berdasarkan urutan	Untuk pengembangan penelitian berikutnya diharapkan hasil ringkasan memiliki urutan berdasarkan sistematika yang baik agar lebih mudah dipahami oleh masyarakat luas.	Penulis menggunakan algoritma yang berbeda serta, serta dalam penelitian ini data yang digunakan berupa teks utuh atau paragraph yang langsung melalui tahap peringkasan sedangkan peringkasan teks yang penulis lakukan adalah berupa dokumen modul dengan hasil peringkasan yang sudah berurutan sesuai bobot topik kalimatnya.

Tabel 2. 2. Matriks *Literatur Review* dan Posisi Penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				bobot		
2	Peringkasan Teks Otomatis pada Modul Pembelajaran Berbahasa Indonesia Menggunakan Metode *Cross Latent Semantic Analysis* (CLSA)	Yunita Maulida Sari, Nenden Siti Fatonah, JEPIN, 2021.	Penerapkan metode Term Frequency-Inverse Document Frequency (TF-IDF) untuk pembobotan kata dan metode Cross *Latent Semantic Analysis* (CLSA) untuk peringkasan teks.	Pengujian akurasi pada peringkasan modul pembelajaran dilakukan dengan cara membandingkan hasil ringkasan manual oleh manusia dan hasil ringkasan sistem. Yang mana pengujian ini menghasilkan rata-rata nilai *fmeasure, precision, dan recall* tertinggi pada *compression rate* 20% dengan nilai berturut-turut 0.3853, 0.432, dan 0.3715.	Kelemahan pada penelitian ini adalah kurang memperhatikan delimiter pada saat melakukan pemecahan kalimat. Karena ada beberapa kata yang menjadi ambigu reque delimiter yang digunakan tanda titik (.) sehingga hasil akurasi sangat rendah.	Penulis menggunakan algoritma yang berbeda dari penelitian ini yaitu LSA serta objek dan jumlah data yang berbeda.
3	Penerapan Algoritma Score-Based Pada Peringkasan Teks Cerpen Otomatis.	Malak Diana D, Bagas Satya D. N, Faisal Rahutomo, SIAP, 2020	Pencarian intisari teks cerpen menggunakan algoritma Score-Based.	Pengujian akurasi peringkasan teks cerpen menggunakan algoritma score-based dilakukan dengan membandingkan ringkasan sistem dengan ringkasan manual dari pakar dengan hasil rata-rata nilai *precision* sebesar 52, *recall* sebesar 44,88889, dan *f-measure*	Kelemahan pada penilitan ini adalah tidak melakukan pembobotan pada masing-masing fitur sehingga tidak diketahui fitur mana yang lebih penting dalam peringkasan teks otomatis cerpen tersebut.	Selain dari algoritma peringkasan yang digunakan berbeda, pada penilitian ini data hanya berisi full teks dengan jumlah kata maksimal 10.000 sedangkan penulis menggunakan data berupa dokumen modul yang berisi huruf, table maupun gambar yang jumlah katanya lebih banyak dari cerpen.

Tabel 2. 2. Matriks *Literatur Review* dan Posisi Penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				sebesar 46,65009. Pengujian kepuasan pembaca dilakukan menggunakan kuisioner dan diperoleh hasil sebesar 71,3%.		
4	Sistem Peringkas Teks Otomatis Multi Dokumen Kliping Artikel Berita Gempa Menggunakan Metode TF-IDF.	Aa Zezen Zainal Abidin, Enung Nurjanah, Jumal Informasi dan Komunikasi STMIK Subang, 2020	Pembuatan prototype system peringkas teks multi dokumen menggunakan metode Term frequency inverse document frequency (TF-IDF).	Aplikasi ini dapat mengimplementasikan peringkas teks otomatis multi dokumen kliping artikel berita di internet metode TF-IDF. Sistem ini dapat membantu mengetahui isi penting dari kliping artikel berita yang banyak di internet. Memiliki akurasi hasil uji responden 54.45% dan uji kemiripan dokumen sebesar 78.023%.	Saran dalam tahapan hasil agar melakukan perbandingan hasil peringkasan dengan aplikasi dan dengan pakar agar tingkat akurasi kebenaran hasil ringkasan lebih di percaya.	Penulis menggunakan metode TF-IDF untuk pembobotan katanya sedangkan dalam peringkasan menggunakan metode LSA namun dalam penelitian ini tahap peringkasan dilakukan cukup hingga tahap pembobotan yang berulang-ulang.
5	Otomatisasi Peringkasan Teks Pada Dokumen Hukum Menggunakan Metode Latent	Imam Fahrur Rozi, Kadek Suarjuna Batubulan dan Millenia Rusbandi, Jurnal	Menguji 10 dokumen hukum berbahasa Indonesia untuk di ringkas dengan	Hasil peringkasan diperoleh untuk compression rate 50 % yaitu *Precision* 54%, *recall* 56%, *f-measure* 55% dan *accuracy* 75%	Pada penelitian ini dokumen yang di ringkas hanya bisa menggunakan format pdf tidak format yang lain.	Peneliti akan melakukan peringkasan teks modul pembelajaran berbahasa Indonesia dengan file format pdf dan doc/docx yang tidak dikunci sedangkan pada

Tabel 2. 2. Matriks *Literatur Review* dan Posisi Penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		Informatika Polinema, 2021	menggunakan algoritma LSA			penelitian ini hanya untuk dokumen berformat pdf
6	Peringkasan Teks Bahasa Indonesia Pada Cerpen Menggunakan Metode Latent Semantic Analysis (LSA)	Nabilah Thahirah Kasim, 2022	Melakukan peringkasan teks cerpen berbahasa indonesia menggunakan Algoritma LSA.	Terdapat 3 tahapan pada peroses peringkasan teks yaitu *Text Processing*, TFIDF dan LSA. data yang digunakan sebanyak 30 teks cerpen dengan hasil pengujian *recall* 68,4%, *precision* 74,93% dan *f-measure* 71,43%.	pada penelitian ini hanya bisa membaca file berformat .txt dan tidak mampu mengolah atau melakukan proses cleaning untuk rumus, simbol dan tabel pada file text. Dokumen berupa file cerpen dengan susunan text pada file sudah rapi melalui seleksi/*editing* manual, sehingga tidak seluruhnya dilakukan oleh sistem.	peneliti melakukan proses preprocessing data secara menyeluruh oleh sistem tanpa ada buntuan tangan manusia dengan dokumen berupa docx dan pdf.
7.	*Comparison of Stemming Test Results of Tala Algorithms with Nazief Adriani in Abstract Documents and National News*	Natalinda Pamungkas, Erika Devi Udayanti, Bonifacius Vicky Indriyono, Wildan Mahmud, Ery Mintorini Erika Norma Wahyu Dorroty	Membandingkan algoritma stemming Nazief-Adriani dan Algoritma Stemming Porter dengan dataset tugas akhir Mahasiswa dan Berita online.	Algoritma Nazief-Adriani lebih unggul di banding algoritma Tala dengan hasil akurasi oleh Nazief-Adriani sebesar 78,47% selama 32,19 detik pada 230 dataset.	Perlu dijelaskan terkait kendala pada algoritma masing-masing yang mempengaruhi hasil akurasi dan kecepatan proses stemming.	Matode stemming Nazief-Adriani pada penelitian tersebut akan peneliti angkat sebagai metode *stemming* dalam penelitian ini akan tetapi hasil metode *stemming* itu akan dikembangkan ke tahap peringkasan teks otomatis.

Tabel 2. 2. Matriks *Literatur Review* dan Posisi Penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		dan Sanina Quamila Putri, Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi, 2023				
8.	*Identification of topics in News Articles Using Algorithm of Porter Stemmer Enhancement and Likelihood Classifier*	Alvida Mustika Rukmi, Devi Andriyani dan Imam Mukhlas, Jurnal of Physics: Conference Series, 2020	Identifikasi topik pada artikel Berita dengan melakukan klasifikasi menggunakan algoritma Likelihood dan Algoritma Stemming Porter.	Berdasarkan hasil pengujian dari 900 data latih dan 90 data uji, diperoleh hasil yang cukup baik akurasi tinggi, yaitu 95,56% untuk klasifikasi kategori dan 97,78% untuk topik identifikasi.	Tidak ada perbandingan dengan penelitian sebelumnya di bidang yang sama.	Tema penelitian berbeda akan tetapi peneliti mengambil referensi algoritma *stemming* yang di gunakan mampu memberikan hasil akurasi tinggi.

2.3. Landasan Teori

2.3.1. Modul Pembelajaran

Modul adalah suatu paket pengajaran yang berisi suatu unit terkecil dan bertahap dari suatu mata pelajaran tertentu. Modul disusun agar mahasiswa dapat menguasai kompetensi yang diajarkan dalam diklat atau kegiatan pembelajaran dengan sebaik-baiknya. Bagi dosen, modul bisa digunakan sebagai acuan dalam menyajikan dan memberikan materi selama diklat atau kegiatan pembelajaran berlangsung.

Modul pembelajaran merupakan sarana pembelajaran yang berisi materi, metode, batasan-batasan materi pembelajaran, petunjuk kegiatan belajar, latihan dan cara mengevaluasi yang dirancang secara sistematis dan menarik untuk mencapai kompetensi yang diharapkan dan dapat digunakan secara mandiri yang memiliki sifat *self contained* artinya dikemas dalam satu kesatuan yang utuh untuk mencapai kompetensi tertentu. Modul juga memiliki sifat membantu dan mendorong pembacanya untuk mampu membelajarkan diri sendiri (*self instructional*) dan tidak bergantung pada media lain (*stand alone*) dalam penggunaannya.

2.3.2. Natural Language Processing (NLP)

NLP adalah cabang dari kecerdasan buatan (*Artficial Intelligence*) yang berhubungan dengan melatih komputer untuk memahami, memproses, dan menghasilkan Bahasa (Rumaisa et al, 2021).

NLP secara khusus menganalisis sintaksis untuk memahami susunan kata pada kalimat agar bisa menangkap makna gramatikal. Menganalisis sintaksis

dilakukan untuk menilai bagaimana bahasa natural sejajar dengan aturan tata bahasa.

Secara umum, ada tahapan dari analisis sintaksis yang sering dipakai, yaitu:

1. *Stemming*: memotong awalan atau akhiran untuk menghilangkan imbuhan, misalnya *bi* dalam *bicycle* dan *er* dalam *lighter*. Pada p
2. *Lemmatization*: pengurangan berbagai bentuk kata untuk memudahkan analisis, misalnya *swim* — *swimming* — *swims* — *swam* adalah berbagai bentuk dari *swim*, maka lemma dari semua kata tersebut adalah *swim*.
3. *Tokenization*: membagi sebagian besar teks berkelanjutan menjadi unit-unit yang berbeda, misalnya: saya mau makan nasi → kalimat tersebut dipecah setiap katanya yaitu saya — mau — makan — nasi, setiap kata ini adalah token.
4. *Parsing*: menganalisis teks menjadi komponen sintaksis logis yang digunakan untuk menguji kesesuaian dengan tata bahasa. *Parsing* contohnya memecah kalimat untuk menjelaskan tiap-tiap elemen. *Parsing* misalnya praktik mengeja, misal B – U → BU, D – I → DI = BUDI.

2.3.3. Peringkasan Teks Otomatis

Peringkasan dokumen adalah proses mengambil teks dari sebuah dokumen, menggali dan menyajikan informasi penting bagi user atau aplikasi dalam bentuk rangkuman yang singkat dan padat. Peringkasan dokumen dapat menjadi solusi bagi setiap orang yang tidak memiliki banyak waktu dan sedang membutuhkan informasi penting dalam tumpukan dokumen yang terus berkembang (Pertiwi, 2022).

Peringkasan teks otomatis atau *Automatic Text Summarization* merupakan proses mengambil dokumen tekstual, mengekstraksi isinya, dan menyajikan konten

yang paling penting untuk pengguna dalam bentuk yang lebih padat dan sesuai dengan kebutuhan pengguna (Rumaisa et al, 2021).

Untuk menentukan frase atau kalimat utama pada text Summarization maka digunakan beberapa fitur yang dijadikan dasar pertimbangan untuk menghitung *weight*, di antaranya:

1. Frekuensi

Kata yang dianggap penting adalah kata yang sering muncul dalam sebuah dokumen. Semakin sering muncul, maka perhitungan skor untuk kata tersebut semakin tinggi. Pengukuran yang umum digunakan untuk menghitung frekuensi kata adalah TF-IDF.

2. Lokasi

Kalimat utama dalam suatu paragraf biasanya terdapat pada bagian awal dan akhir dari sebuah paragraf, sehingga kalimat ini memiliki kesempatan yang lebih besar untuk diikutsertakan dalam sebuah ringkasan daripada kalimat yang berada di tengah paragraph.

3. Cue *Method*

Pentingnya suatu ide biasanya tersirat dari kalimat: "*in summary*", "*in conclusion*", "*the paper describes*", atau "kesimpulannya adalah", "ringkasannya".

4. Judul/Kepala Berita

Kata yang ada pada judul dan kepala/pokok berita besar kemungkinannya berhubungan dengan ringkasan. Kata-kata yang ada pada sebuah judul juga mengindikasikan topik dari suatu dokumen.

5. Panjang Kalimat

Pada umumnya, kalimat yang terlalu panjang ataupun pendek tidak cocok digunakan dalam sebuah ringkasan

6. Kemiripan

Kemiripan dapat dikalkulasi dengan pengetahuan linguistik. Hal ini mengindikasi kemiripan kalimat yang digunakan dalam judul dan dalam isi dokumen.

7. Kata Benda

Penggunaan kata benda yang tepat harus diperhatikan. Ringkasan harus menggunakan kata benda yang tepat, misalnya nama seseorang, nama tempat ataupun organisasi.

8. Kedekatan

Jarak antar kata dalam sebuah *entity* menjadi sebuah faktor untuk membuat relasi antar *entity*.

2.3.4. *Latent Semantic Analysis* (LSA)

Latent Semantic Analysis (LSA) atau disebut juga dengan *Latent Semantic Indexing* (LSI), merupakan sebuah metode *automatic indexing* dan *retrival* yang memanfaatkan banyaknya kata yang dicari pada sebuah dokumen *semantic structure* (struktur asosiasi term dengan dokumen) yang secara implisit terdapat dalam suatu dokumen untuk digunakan dalam pencarian dokumen yang relevan dengan term dalam query (Wahyuni et al, 2021).

LSA bisa digunakan untuk menilai esai dengan mengkonversikan esai menjadi matriks-matriks yang diberi nilai pada masing-masing term untuk dicari

kesamaan dengan term referensi. Secara umum, langkah-langkah LSA dalam penilaian esai adalah sebagai berikut:

1. *Text Preprocessing*

Preprocessing adalah proses normalisasi teks sehingga informasi yang dimuat merupakan bagian yang padat dan ringkas namun tetap merepresentasikan informasi yang termuat didalamnya. Dalam tahap ini, terdapat beberapa proses diantaranya:

a. *Stopwords Removal* : Pada *stopwords removal*, kata-kata yang tergolong sebagai kata depan, kata penghubung, dan kata-kata lain yang tidak mewakili makna dari kalimat akan dieliminasi. Contohnya adalah sebagai berikut:
   - Menghapus kata-kata "*am, is, are, and, in, etc*" dan berbagai singkatan. Apabila dalam konteks Bahasa Indonesia, contoh kata-kata yang dihapus adalah "yang, di, ke, dari, pada, dalam, dan".
   - Mengubah kata yang diawali dengan huruf besar menjadi huruf kecil.

b. *Stemming* : Langkah berikutnya adalah *stemming*. Pada proses ini, kata akan dinormalkan menjadi kata dasar pembentuk kata tersebut. Caranya adalah dengan menghilangkan imbuhan yang melekat pada kata, sehingga hasilnya adalah kata dasarnya. Apabila dalam Bahasa Inggris, proses *stemming* bisa mengikutsertakan pengembalian bentuk tense dari kata kerja bentuk ke-2 atau ke-3 menjadi kata kerja bentuk ke-1.

2. *Term-document Matrix*

Setelah melalui stopwords removal dan *stemming*, matriks term-document dibangun dengan menempatkan kata hasil proses *stemming* (*term*) ke dalam baris. Matriks ini disebut *term-document matrix*. Setiap baris mewakili sebuah kata yang unik, sedangkan setiap kolom mewakili konteks dari mana kata-kata tersebut diambil. Konteks yang dimaksud bisa berupa kalimat, paragraf, atau seluruh bagian dari teks. Di bawah ini merupakan contoh *term-document matrix*:

Tabel 2. 3. *Term Document Matrix*

	Document 1	Document 2	Document 3	Document n
Term 1	1	2	0	N
Term 2	1	0	3	N
Term 3	1	1	0	N
Term 4	1	0	0	N
Term 5	0	0	4	N
Term 6	1	1	0	N
Term 7	1	0	0	N
Term 8	0	2	1	N
Term 9	1	1	0	N
Term n	n	n	n	N

Pada tabel di atas, baris pertama mewakili *stemmed term* (*term* 1, *term* 2, dst), dan bagian kolom mewakili konteks, yaitu teks. Nilai yang terletak pada setiap *cell* pada tabel menunjukkan berapa kali sebuah *term* muncul dalam sebuah dokumen. Contohnya, *term* 1 muncul 1 kali pada dokumen ke-1, dan muncul 2 kali pada dokumen ke-2, namun term 1 tidak muncul pada dokumen 3, dan seterusnya.

3. *Singular Value Decomposition*

Singular Value Decomposition (SVD) adalah salah satu teknik reduksi dimensi yang bermanfaat untuk memperkecil nilai kompleksitas dalam pemrosesan

term-document matrix. SVD merupakan teorema aljabar linier yang menyebutkan bahwa persegi panjang dari *term-document matrix* dapat dipecah/didekomposisikan menjadi tiga matriks, yaitu :

a. Matriks ortogonal U

b. Matriks diagonal S

c. Transpose dari matriks ortogonal V

Yang dirumuskan dengan :

$A_{mn} = U_{mm} \times S_{mn} \times V^T_{nn}$

A_{mn} = Matriks Awal

U_{mm} = Matriks Ortogonal U

S_{mn} = matriks diagonal S

V^T_{nn} = transpose matriks orthogonal V

Hasil dari proses SVD adalah vektor yang akan digunakan untuk menghitung similaritasnya dengan pendekatan *cosine similarity*.

4. *Cosine Similarity Measurement*

Cosine similarity digunakan untuk menghitung nilai kosinus sudut antara vektor dokumen dengan vektor kueri. Semakin kecil sudut yang dihasilkan, maka tingkat kemiripan esai semakin tinggi. Formula dari *cosine similarity* adalah sebagai berikut:

$$\cos \alpha = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{|\boldsymbol{A}|\,|\boldsymbol{B}|} = \frac{\sum_{i}^{n}=_1 \boldsymbol{A}_i \, X \, \boldsymbol{B}_i}{\sqrt{\sum_{i=1}^{n}(\boldsymbol{A}_i)^2} \, X \, \sqrt{\sum_{i=1}^{n}(\boldsymbol{B}_i)^2}}$$

A = Vektor dokumen

B = Vektor Kueri

A•B = Perkalian *dot* vektor A dan vektor B

|A| = Panjang vektor A

|B| = Panjang vector B

|A| |B| = *Cross product* antara |A| dan |B|

α = Sudut yang terbentuk antara vektor A dan B

Dari hasil *cosine similarity*, akan didapatkan nilai yang akan dibandingkan dengan penilaian manusia untuk diuji selisih nilainya.

2.3.5. *ROUGE*

ROUGE atau *Recall-Oriented Understudy for Gisting Evaluation*, adalah seperangkat metrik dan paket perangkat lunak yang digunakan untuk mengevaluasi peringkasan otomatis dan perangkat lunak terjemahan mesin dalam pemrosesan bahasa alami. Metrik membandingkan ringkasan atau terjemahan yang dihasilkan secara otomatis dengan referensi atau serangkaian ringkasan atau terjemahan referensi (diproduksi manusia). (Halimah et al., 2022).

Skor ROUGE bercabang menjadi skor ROUGE-1, ROUGE-2, dan ROUGE-L. ROUGE-1 *Precision* and *Recall* membandingkan kesamaan uni-gram antara ringkasan referensi dan kandidat. Dengan uni-gram, yang di maksud hanyalah setiap token perbandingan adalah satu kata. ROUGE-2 *Precision* and *Recall* membandingkan kesamaan bi-gram antara ringkasan referensi dan kandidat. Yang di maksud dengan bi-gram adalah setiap token perbandingan adalah 2 kata berurutan dari referensi dan ringkasan kandidat. Sedangkan ROUGE-L *Precision* and *Recall* mengukur kata *Longest Common Subsequence* (LCS) antara ringkasan referensi dan kandidat. Dengan LCS merujuk pada token kata yang berurutan, tetapi belum tentu harus berurutan.

ROUGE-1, ROUGE-2 dan ROUGE-L *Precision/Recall* memberikan representasi yang baik tentang seberapa akurat ringkasan yang dihasilkan model mewakili ringkasan beranotasi emas. Untuk membuat skor lebih ringkas, biasanya skor F1, yang merupakan rata-rata harmonik antara *Precision* dan *Recall*, dihitung untuk semua skor ROUGE. Berikut persamaan untuk metode ROUGE dalam mencari nilai recall, precision dan fmeasure.

$$ROUGE - 1\ recall = \frac{jumlah\ unigram\ kata\ yang\ sama}{Total\ kata\ di\ ringkasan\ manual}$$

$$ROUGE - 1\ precision = \frac{jumlah\ unigram\ kata\ yang\ sama}{Total\ kata\ di\ ringkasan\ sistem}$$

$$ROUGE - 2\ recall = \frac{jumlah\ bigram\ kata\ yang\ sama}{Total\ kata\ di\ ringkasan\ manual}$$

$$ROUGE - 2\ precision = \frac{jumlah\ bigram\ kata\ yang\ sama}{Total\ kata\ di\ ringkasan\ sistem}$$

$$ROUGE - L\ recall = \frac{LCS\ (sistem, manual)}{Total\ kata\ di\ ringkasan\ manual}$$

$$ROUGE - L\ precision = \frac{LCS\ (sistem, manual)}{Total\ kata\ di\ ringkasan\ sistem}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian ini adalah deskriptif, jenis ini bertujuan membuat deskripsi secara sistematis, faktual, dan akurat tentang fakta-fakta dan sifat-sifat populasi atau objek tertentu. Penelitian deskriptif adalah penelitian yang dimaksudkan untuk menyelidiki keadaan, kondisi atau hal-hal yang sudah disebutkan, yang hasilnya dipaparkan dalam bentuk laporan penelitian (Kue et al, 2022). Desain analisis ini tidak dimaksudkan untuk menguji suatu hipotesis tertentu, atau menguji hubungan antar variabel.

Dalam penelitian ini, peneliti menggunakan metodologi kuantitatif dimana riset kuantitatif adalah riset yang menggambarkan atau menjelaskan suatu masalah yang hasilnya dapat digeneralisasikan. Dalam hal ini tidak terlalu mementingkan kedalaman data atau analisis akan tetapi lebih mementingkan aspek keluasan data atau hasil riset dianggap merupakan representasi dari seluruh populasi

3.2. Metode Pengumpulan Data

Metode yang digunakan dalam penelitian ini adalah studi dokumen dimana penulis mendapatkan dataset untuk penelitian ini secara manual. Pada penelitian ini menggunakan data sebanyak 25 file modul pembelajaran yang berasal dari modul para dosen Universitas Teknologi Mataram yang tidak dikunci atau lock, dengan format .docx dan format .pdf.

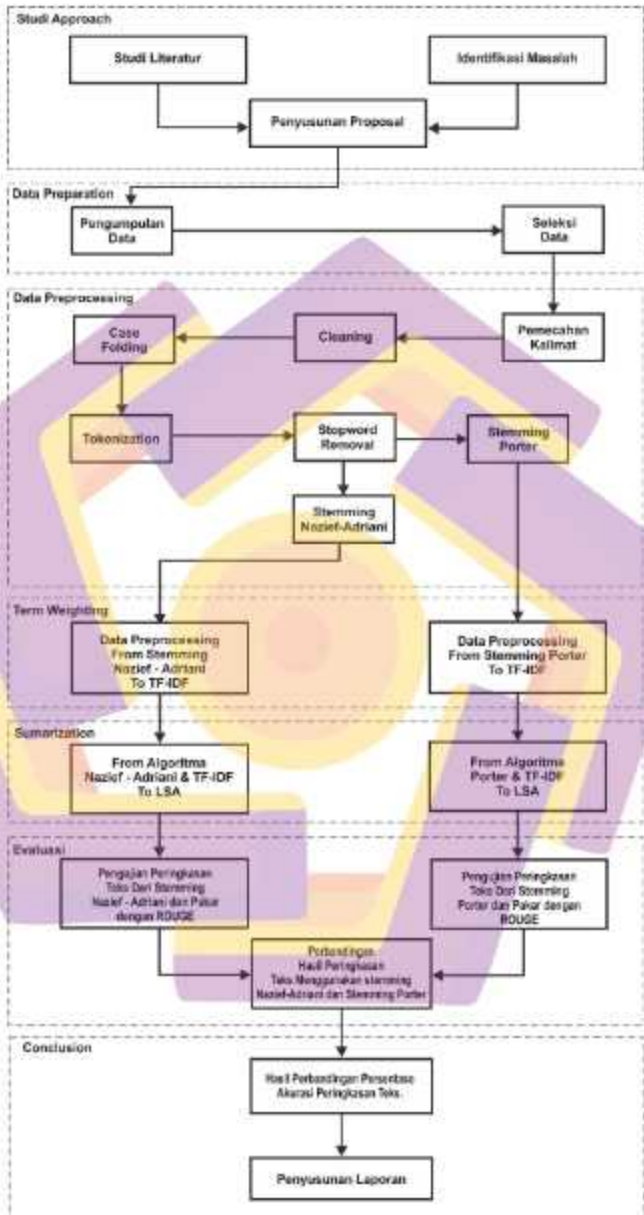
Keseluruh datanya akan di ringkas oleh sistem dan dilakukan pengujian untuk dibandingkan oleh data hasil peringkasan manual dari ke 25 modul tersebut oleh pakar.

3.3. Metode Analisis Data

Setelah dilakukan tahapan pengumpulan data selanjutnya adalah analisis data pada penelitian ini akan dilakukan pada tahapan *preprocessing* dimana pada tahapan ini merupakan suatu awal tahap yang perlu dilakukan sebelum meringkas suatu teks. Tahapan dalam *preprocessing* ini adalah *cleaning*, pemecahan kalimat, *case folding*, *stemming* menggunakan Algoritma Nazief -Adriani dan Algoritma Porter, *tokenization*, dan *stopword removal*. Hasil analisis data pada tahapan *preprocessing* ini adalah berupa susunan kata dasar yang dihasilkan dari dua metode *stemming* dan selanjutnya data tersebut akan menuju ke tahapan pembobotan kata dengan metode TF-IDF (*Term frequency-Inverse document frequency*) untuk menghasilkan bobot dari setiap kata dasar yang akan digunakan pada proses peringkasan teks menggunakan algoritma LSA

3.4. Alur Penelitian

Pada penelitian ini terdapat beberapa tahapan penelitian diantaranya; pendekatan penelitian, persiapan data, *preprocessing*, pembobotan kata, peringkasan teks lalu evaluasi dan kesimpulan. Tahapan alur penelitian ditujukan pada Gambar 3.1 sebagai berikut :



Gambar 3. 1. Alur Penelitian

1. Pendekatan Penelitian / *Studi Approach*

Tahapan pertama pada penelitian ini adalah melakukan studi literatur dan identifikasi masalah untuk penyusunan laporan proposal penelitian.

2. Persiapan Data / *Data Preparation*

Pada tahapan ini akan melakukan pengumpulan data yang di dapatkan dari dosen Universitas Teknologi Mataram yaitu sejumlah 25 modul pembelajaran dengan format pdf dan docx, selanjutnya data tersebut akan di seleksi yang berbahasa Indonesia dan modul pembelajaran teori.

3. *Data Preprocessing*

Tahapan preprocessing dalam penelitian ini mencakup 6 proses yaitu dimulai dengan pemecahan kalimat lalu *cleaning*, *case folding*, *stemming*, *tokenization* dan yang terakhir *stopword removal*. Untuk penjelasan lebih lanjut sebagai berikut:

a. Pemecahan Kalimat: Pada tahapan pemecahan kalimat yang berarti memecah dokumen yang telah diinputkan menjadi per kalimat. Pemisah (*delimiter*) yang digunakan yaitu (. , ?). Hasil dari pemecahan kalimat selanjutnya dapat digunakan untuk tahapan selanjutnya.

b. *Cleaning*: Pada tahap ini kumpulan kalimat yang telah dipecah, selanjutnya dilakukan proses cleaning yang bertujuan untuk membersihkan *noise* yang ada dalam teks. Beberapa contoh *noise* yang akan dibersihkan yaitu: angka, tanda buka kurung, dan lain-lain.

c. *Case Folding*: Setelah melalui tahap *cleaning*, selanjutnya yaitu tahapan *case folding*. Tahap ini berfungsi untuk mengganti huruf kapital (*uppercase*) menjadi

huruf kecil (*lowercase*). Sehingga teks menjadi sama rata dan tidak ditemukan lagi huruf kapital (*uppercase*) dalam data.

d. *Tokenization*: Tahap ini berfungsi untuk memotong atau memisahkan atau memecah kalimat menjadi perkata berdasarkan spasi sebagai pemotong/ pemisah/ pemecah kata tersebut.

e. *Stopword Removal*: Dalam tahap ini kata-kata yang dianggap tidak penting akan dilakukan penghapusan, yang bertujuan untuk mengurangi jumlah kata yang akan diproses. Contoh kata yang akan dihilangkan seperti; "dan", "atau", "dia", "ia", "adalah", "dari", dan lain-lain.

f. *Stemming*: Tahapan yang terakhir pada preprocessing adalah tahapan yang bertujuan untuk mengubah kata berimbuhan menjadi kata dasar. Contohnya seperti pada kata "membeli" diganti menjadi "beli", kata "berlibur" diganti menjadi "libur". Dalam penelitian ini menggunakan dua aturan *stemming* yang berbeda antara lain *stemming* yang berbasis kamus yaitu algoritma Nazief-Adriani dengan *stemming* berbasis aturan imbuhan yaitu algoritma Porter.

4. Pembobotan Kata / *Term Weighting*

Term frequency-Inverse document frequency (TF-IDF): merupakan ekstraksi kalimat dengan cara memberikan nilai atau bobot pada kata. TF-IDF ialah salah satu metode perhitungan bobot kata dengan cara mengekstraksi ciri suatu teks. Proses perhitungan TF-IDF dilakukan agar mendapatkan bobot kata yang terdapat pada suatu dokumen. Semakin sering kata tersebut muncul pada sebuah dokumen maka nilainya semakin besar. Bobot kata akan memperhitungkan kebalikan frekuensi dokumen yang terdapat sebuah kata (*Inverse Document Frequency*).

Proses TF-IDF dapat dituliskan seperti pada rumus 1 dan rumus 2:

(Rumus 1)

$$W = TF * IDF$$

Keterangan:

W : bobot dokumen ke-d terhadap kata ke-t (nilaibobot dari setiap kata pada sebuah dokumen).

TF : jumlah kata yang dihitung pada kalimat (D)

IDF : Inverse Document Frequency

Untuk nilai IDF dapat dicari dengan rumus 2:

(Rumus 2)

$$IDF = \text{Log}(\frac{N}{DF})$$

Keterangan:

IDF : banyaknya dokumen yang mengandung kata (Document Frequency) dengan jumlah kalimat (D)

N : total kalimat dalam dokumen

DF : banyaknya kalimat yang mengandung kata.

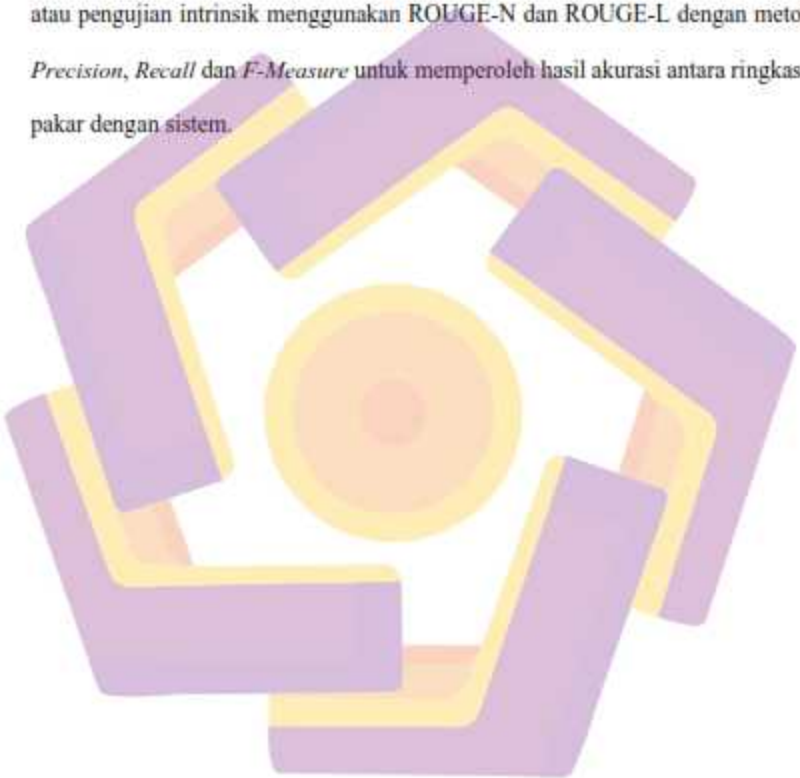
5. Peringkasan / *Summarization*

Setelah mendapatkan hasil pembobotan dari setiap kata dengan menggunakan algoritma TF-IDF. Maka tahap selanjutnya adalah pemilihan kalimat Ringkasan dengan menggunakan algoritma *Latent Semantic Analysis* (LSA). Cara kerja LSA ialah dengan menghasilkan sebuah model yang didapat dengan mencatat kemunculan-kemunculan kata dari tiap-tiap dokumen yang direpresentasikan dalam sebuah matrik *term-document*, setelah itu dilakukan proses *Singular Value*

Decomposition (SVD) yang akan digunakan untuk mendapatkan *Cosine Similarity* (nilai kemiripan) antara satu dokumen dengan dokumen yang lain.

6. Evaluasi

Penulis melakukan penelitian ini dengan menggunakan metode evaluasi atau pengujian intrinsik menggunakan ROUGE-N dan ROUGE-L dengan metode *Precision*, *Recall* dan *F-Measure* untuk memperoleh hasil akurasi antara ringkasan pakar dengan sistem.



BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Persiapan Data

Persiapan data merupakan tahapan pertama pada peringkasan modul pembelajaran ini. Pada tahapan persiapan data, ada dua langkah yang dilakukan yaitu pengumpulan data dan seleksi data.

4.1.1. Pengumpulan Data

Dataset berupa modul yang berasal dari Universitas Teknologi Mataram. Modul dikumpulkan secara manual dari Dekan masing-masing fakultas. Selain data berupa modul pembelajaran, dataset yang dikumpulkan juga berupa data hasil dari peringkasan oleh Pakar yang mana hasil peringkasan oleh pakar dapat dilihat pada link ini: https://bit.ly/peringkasanpakar.

Pakar pada penelitian ini adalah Bapak Dr. Muhammad Multazam, S.Kom., M.Kom selaku dosen dengan jabatan struktural Dekan pada Fakultas Teknologi dan Informasi (FTIK) dan Ibu Dr. Henni Comala Hikmi, S.E., M.Pd selaku dosen dengan jabatan struktural Dekan Fakultas Bisnis dan Hukum (FBH) serta Bapak Lalu Wirajayadi, S.Pd., M.Pd selaku dosen Bahasa Indonesia dan Sastra pada Universitas Teknologi Mataram.

4.1.1. Seleksi Data

Jumlah modul yang di dapatkan dari Fakults Bisnis dan Hukum (FBH) sebanyak 102 judul modul dan dari Fakultas Teknologi dan Informasi Komputer (FTIK) sebanyak 197 judul modul. Penelitian ini mengambil 20 modul dari FBH

dan 20 modul dari FTIK. Berikut daftar modul yang digunakan dalam penelitian ini pada tabel 4 .

Tabel 4. 1. Daftar Dataset

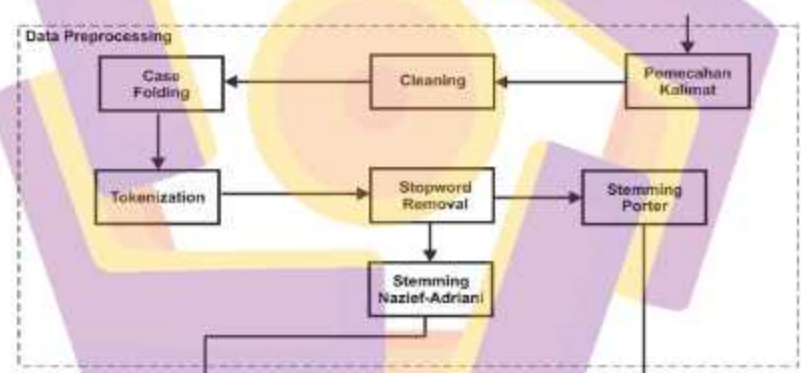
No	Judul Modul	Jumlah Kata
1	Modul Sistem Operasi.docx	8.831
2	Modul Technopreneurship I.docx	8.952
3	Modul Kecakapan Antarpersonal.pdf	28.905
4	Modul Technopreneurship II.pdf	7.213
5	Modul Manajemen operasi perkantoran.docx	6.765
6	Modul Efisiensi Kantor.Docx	6.262
7	Modul Entrepreneurship 2.Docx	7.213
8	Modul KNB Indonesia.Docx	7.062
9	Modul Komunikasi Niaga.Docx	5.991
10	Modul Manajemen Kearsipan.Docx	11.355
11	Modul Manajemen Organisasi Dan Produksi.Docx	8.492
12	Modul Pengantar Ilmu Administrasi.Docx	6.195
13	Modul Sistem Infromasi Manajemen.Docx	13.143
14	Modul Statistik Deskriptif.Docx	7.154
15	Modul Analisis Laporan Keuangan.Pdf	16.369
16	Modul Manajemen Perkantoran 2020.Pdf	23.003
17	Modul Metodologi Penelitian Bisnis.Pdf	27.311
18	Modul Penganggaran Perusahaan.Pdf	17.992
19	Modul Statistik 1.Pdf	16.064
20	Modul Statistik 2.Pdf	28.679
21	Modul Studi Kelayakan Bisnis.Pdf	27.709
22	Modul Surat - Menyurat.Pdf	15.009
23	Modul Umkm Dan Koperasi.Pdf	18.649
24	Modul-Perpajakan.Pdf	18.674
25	Manajemen Organisai Perkantoran.pdf	15.725
Rata-Rata		**14.349**

Tabel 4.1 berisi judul modul dan ekstensi file modul serta jumlah kata yang terkandung dalam setiap modul yang selanjutnya akan di lakukan peringkasan otomatis. Total modul yang digunakan adalah 25 judul dengan rata-rata jumlah 14.349 kata.

4.2. Data Preprocessing

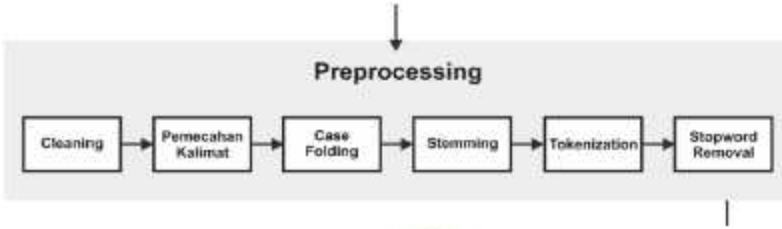
Terdapat 6 proses pada tahapan *preprocessing* dimana hasil akhir pada tahapan *preprocessing* ini adalah susunan kata dasar dari modul yang akan di ringkas dalam bentuk array.

Tahapan *preprocessing* pada penelitian seperti pada gambar 4.1.



Gambar 4. 1. Tahapan *Preprocessing*

Urutan dalam tahapan *preprocessing* ini berbeda dari penelitian sebelumnya oleh Yunita Maulidia Sari dan Nenden Sri Fatonah pada penelitian dengan judu "Peringkasan Teks Otomatis pada Modul Pembelajaran Berbahasa Indonesia Menggunakan Metode *Cross Latent Semantic Analysis* (CLSA)"seperti pada Gambar 4.2.



Gambar 4. 2. Tahapan *Preprocessing* Penelitian Terdahulu.

Urutan tahapan *preprocessing* seperti pada gambar 4.1 oleh penelitian terdahulu Sari & Fatonah, 2021, mampu memberikan hasil lebih baik pada peringkasan teks karena tahapan pemecahan kalimat sebagai tahapan pertama akan membuat dokumen dalam bentuk *array* dengan memisahkannya menjadi perkalimat sehingga jika selanjutnya melalui tahap *Cleaning*, makna pembahasan dalam dokumen tidak terputus karena masih dalam satu kalimat utuh. Setelah melalui tahapan *Case Folding* dan pemecahan kalimat menjadi perkata (*Word Tokenization*) yang hasilnya kata dalam bentuk *array* untuk memudahkan proses *Stopword Removal* atau penghapusan kata bantu yang tidak memiliki makna penting dalam dokumen agar tidak masuk ke tahapan stemming. Pentingnya dilakukan *Stopword Removal* terlebih dahulu sebelum stemming bertujuan agar proses *stemming* tidak berlangsung lama karena kata yang di stemming hanya kata-kata penting pada dokumen.

Tahapan *Preprocessing* untuk lebih jelasnya seperti berikut:

4.2.1. Pemecahan Kalimat

Tahapan pertama adalah pemecahan kalimat, karena dataset modul yang digunakan pada penelitian ini berupa dokumen tunggal sehingga harus di ekstrak

dari format asalnya yaitu pdf atau docx menjadi susunan teks. Pada penelitian ini menggunakan *library* python Aspose.Words seperti pada gambar 4.3.

```
#Ekstrak PDF/DOCX to Text
!pip install aspose-words
import aspose.words as aw
doc = aw.Document(open_filename)
DOCUMENT=doc.to_string(aw.SaveFormat.TEXT)
print(DOCUMENT)
```

Gambar 4. 3. Ekstraksi pdf/docx to text

Gambar 4.3 menjelaskan proses ekstraksi dokumen menjadi bentuk teks atau susunan kata dalam bentuk paragraf. Selanjutnya teks tersebut akan di pisah per kalimat dengan menggunakan tanda (. , ?) sebagai pemisah kalimat.

```
sentences = nltk.sent_tokenize(DOCUMENT)
print (sentences)
```

Gambar 4. 4. Proses Pemisahan Kalimat

Gambar 4.4 menjelaskan bahwa pada tahapan pemisahan kata disini menggunakan *library nltk.sent_tokenize* sehingga hasil dari tahapan ini berupa kalimat dalam bentuk array.

4.2.2. *Cleaning*

Pada tahapan cleaning akan dilakukan proses penghapusan atau membersihkan *noise* yang ada dalam kalimat seperti pada gambar 4. 5.

```
# menghilangkan yang bukan huruf
doc = re.sub(r'[^a-zA-Z\s]', '', doc, re.I|re.A)
# menghilangkan url
doc = re.sub('((www\.[^\s]+)|(https?://[^\s]+)|(http?://[^\s]+))',' ',doc, flags=re.MULTILINE)
# menghilangkan mention, link, hastag
doc = ' '.join(re.sub("([@#][A-Za-z0-9]+)|([\w+:\/\/\S+)"," ", doc).split())
#menghilangkan karakter byte (b')
doc = re.sub(r'(b\'{1,2})',"", doc)
# menghilangkan digit angka
doc = re.sub(r'\d+', '', doc)
#menghilangkan tanda baca
doc = doc.translate(str.maketrans("","",string.punctuation))
```

Gambar 4. 5. Proses *Cleaning*

Gambar 4.5 menjelaskan proses *cleaning* teks pada penelitian ini meliputi penghapusan yang tidak termasuk huruf, penghapusan url, mention dan hashtag, menghilangkan karakter byte, menghilangkan angka dan menghilangkan tanda baca pada teks.

4.2.3. *Case Folding*

Setelah melalui tahap *cleaning*, selanjutnya adalah tahapan *Case Folding* yaitu tahapan yang berfungsi untuk mengganti huruf kapital menjadi huruf kecil semua sehingga teks menjadi sama rata seperti pada gambar 4.6.

```
#merubah menjadi huruf kecil semua
doc = doc.lower()
doc = doc.strip()
```

Gambar 4. 6. Proses *Case Folding*

4.2.4. *Tokenization*

Tahapan *tokenization* adalah tahapan yang hampir mirip dengan tahapan pemecahan kalimat dimana pada tahapan ini akan memecah kalimat menjadi kata dengan menggunakan spasi sebagai pemisah. Pada tahapan ini peneliti menggunakan *library nltk.word_tokenize*.

4.2.5. *Stopword Removal*

Tahapan *Stopword Removal* adalah proses penghapusan kata yang di anggap tidak penting seperti kata penghubung "dan","atau","BAB I" dan lain-lain seperti pada gambar 4.7.

```
#data stopwords dari NLTK Corpus
from nltk.corpus import stopwords
stopwordsnltkid = set(stopwords.words('indonesian'))
stopwordsnltken = set(stopwords.words('english'))

#data stopwords dataset manual
manual = '/content/drive/MyDrive/Tesis/stopwords_id.txt'
f = open(manual, 'r')
stopwordsmanual = []
for line in f:
    stripped_line = line.strip()
    line_list = stripped_line.split()
    stopwordsmanual.append(line_list[0])

#menggabungkan 3 sumber data stopwords
all_stopwords =set.union(stopwordsnltkid,stopwordsnltken,stopwordsmanual)
```

Gambar 4. 7. *Stopword Removal.*

Gambar 4.7 menjelaskan dasar penghapusan kata pada penelitian ini adalah file stopwords_id.txt yang berisi 879 kata di unduh melalui link *https://www.kaggle.com/* dan penambahan kata secara manual oleh peneliti disesuaikan dengan kebutuhan modul pembelajaran serta stopwords dari *library nltk.corpus* yang berbahasa indonesia dan berbahasa inggris.

4.2.6. Stemming

Tahapan terakhir *preprocessing* pada penelitian ini adalah tahap stemming dimana pada tahapan ini akan mengolah kata-kata inti menjadi kata dasar untuk selanjutnya akan dihitung bobot kemunculan kata dasar tersebut pada modul.

Peneliti menggunakan dua algoritma stemming yaitu algoritma Nazief & Adriani dan algoritma Porter.

a. Algoritma Nazief & Adriani

Tahapan ini membuang semua imbuhan yang terdapat pada suatu kata. Tahapan ini menerapkan library Sastrawi yang mengimplementasikan algortima

Nazief Adriani (Ismi & Ardianto, 2020). Seperti pada gambar 4.8 dimana pada gambar tersebut menjelaskan proses perulangan untuk melakukan stemming Sastrawi pada setiap kata dalam teks.

```
#stemming Nazief-Adriani menggunakan library PySastrawi
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

factory = StemmerFactory()
stemmer = factory.create_stemmer()
#norm_sentences = stemmer.stem(norm_sentences)

norm_sentences=[]
for kata in prepro_sentences:
  norm_sentences.append(stemmer.stem(kata))
print (norm_sentences)
```

Gambar 4. 8. *Stemming* Nazief & Adriani

Algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut:

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tesebut adalah *root word*. Maka algoritma berhenti.
2. *Inflection Suffixes* ("-lah", "-kah", "-ku", "-mu", atau "-nya") dibuang. Jika berupa *particles* ("-lah", "-kah", "-tah" atau "-pun") maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* ("-ku", "-mu", atau "-nya"), jika ada.
3. Hapus *Derivation Suffixes* ("-i", "-an" atau "-kan"). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
   a. Jika "-an" telah dihapus dan huruf terakhir dari kata tersebut adalah "-k", maka "-k" juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.

b. Akhiran yang dihapus ("-i", "-an" atau "-kan") dikembalikan, lanjut ke langkah 4.

4. Hapus *Derivation Prefix*. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
   a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
   b. For i = 1 to 3, tentukan tipe awalan kemudian hapus awalan. Jika *root word* belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Melakukan *Recoding*.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

Tipe awalan ditentukan melalui langkah-langkah berikut:

1. Jika awalannya adalah: "di-", "ke-", atau "se-" maka tipe awalannya secara berturut-turut adalah "di-", "ke-", atau "se-".
2. Jika awalannya adalah "te-", "me-", "be-", atau "pe-" maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.
   3. Jika dua karakter pertama bukan "di-", "ke-", "se-", "te-", "be-", "me-", atau "pe-" maka berhenti.

Tabel 4. 2. Kombinasi Awalan Akhiran yang diizinkan

Awalan	Akhiran yang tidak di izinkan
be-	-i

Tabel 4. 3. Kombinasi Awalan Akhiran yang diizinkan (Lanjutan)

di-	-an
ke-	-i, -kan
me-	-an
se-	-i,-kan

Untuk mengatasi keterbatasan pada algoritma di atas, maka ditambahkan aturan-aturan dibawah ini:

1. Aturan untuk reduplikasi
   a. Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka *root word* adalah bentuk tunggalnya, contoh : "buku-buku" *root word*-nya adalah "buku".
   b. Kata lain, misalnya "bolak-balik", "berbalas-balasan, dan "seolah-olah". Untuk mendapatkan root word-nya, kedua kata diartikan secara terpisah. Jika keduanya memiliki *root word* yang sama maka diubah menjadi bentuk tunggal, contoh: kata "berbalas-balasan", "berbalas" dan "balasan" memiliki *root word* yang sama yaitu "balas", *maka root word* "berbalas-balasan" adalah "balas". Sebaliknya, pada kata "bolak-balik", "bolak" dan "balik" memiliki *root word* yang berbeda, maka *root word*-nya adalah "bolak-balik"
2. Tambahan bentuk awalan dan akhiran serta aturannya.
   a. Untuk tipe awalan "mem-", kata yang diawali dengan awalan "memp-" memiliki tipe awalan "mem-".

b. Tipe awalan "meng-", kata yang diawali dengan awalan "mengk-" memiliki tipe awalan "meng-".

Berikut hasil stemming kata pada penggalan kalimat di modul pembelajaran Sistem Infromasi Manajemen.docx.

Tabel 4. 4. *Stemming* Nazief-Adriani

Isi Paragraf Asli	Hasil preprocessing menggunakan stemming Nazief-Adriani
Sistem informasi membantu perusahaan menyajikan laporan keuangan dalam bentuk informasi yang akurat dan terpercaya dengan memanfaatkan sistem informasi akuntansi untuk mencapai keunggulan perusahaan.	sistem informasi bantu usaha saji lapor uang bentuk informasi akurat percaya manfaat sistem informasi akuntansi capai unggul usaha

b. Algoritma Porter

Algoritma kedua yang digunakan dalam penelitian ini adalah Algoritma Porter. Adapun langkah-langkah algoritma ini adalah sebagai berikut:

1. Hapus *Particle*
2. Hapus *Possesive Pronoun*.
3. Hapus awalan pertama. Jika tidak ada lanjutkan ke langkah 4a, jika ada cari maka lanjutkan ke langkah 4b.
4. a. Hapus awalan kedua, lanjutkan ke langkah 5a.

   b. Hapus akhiran, jika tidak ditemukan maka kata tersebut diasumsikan sebagai *root word*. Jika ditemukan maka lanjutkan ke langkah 5b.
5. a. Hapus akhiran. Kemudian kata akhir diasumsikan sebagai *root word*

   b. Hapus awalan kedua. Kemudian kata akhir diasumsikan sebagai *root word*.

Terdapat 5 kelompok aturan pada Algoritma Porter untuk Bahasa indonesia ini. Aturan tersebut dapat dilihat pada Tabel 4.4 sampai Tabel 4.8.

Tabel 4. 5. Aturan Untuk *Inflectional Particle*

Akhiran	Replacement	Measure Condition	Additional Condition	Contoh
-kah	NULL	2	NULL	bukukah
-lah	NULL	2	NULL	pergilah
-pun	NULL	2	NULL	apapun

Tabel 4. 6. Aturan Untuk *Inflectional Possesive Pronoun*

Akhiran	Replacement	Measure Condition	Additional Condition	Contoh
-ku	NULL	2	NULL	bukuku
-mu	NULL	2	NULL	bukumu
-nya	NULL	2	NULL	bukunya

Tabel 4. 7. Aturan Untuk *First Order Derivational Prefix*

Awalan	Replacement	Measure Condition	Additional Condition	Contoh
meng-	NULL	2	NULL	mengukur →ukur
meny-	S	2	V….*	menyapu →sapu
men-	NULL	2	NULL	menduga →duga
mem-	P	2	V…	memaksa →paksa
mem-	NULL	2	NULL	membaca→baca
me-	NULL	2	NULL	merusak →rusak
peng-	NULL	2	NULL	pengukur→ ukur

Tabel 4. 8. Aturan Untuk *First Order Derivational Prefix* (Lanjutan)

peny-	S	2	V…	penyapu→sapu
pen-	NULL	2	NULL	penduga→duga
pem-	P	2	V…	pemaksa→paksa
pem-	NULL	2	NULL	pembaca-baca
di-	NULL	2	NULL	diukur→ukur
ter-	NULL	2	NULL	tersapu→sapu
ke-	NULL	2	NULL	kekasih→kasih

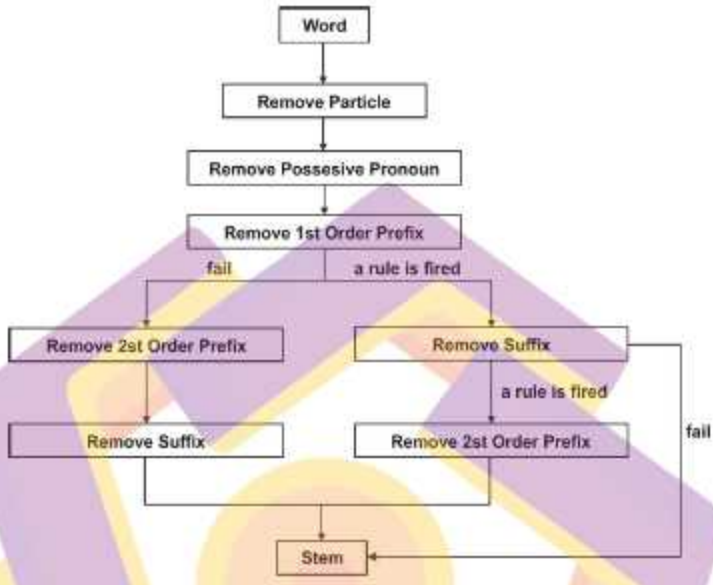
Tabel 4. 9. Aturan untuk *Second Order Derivational* Prefix

Awalan	Replacement	Measure Condition	Additional Condition	Contoh
ber-	NULL	2	NULL	berlari→lari
bel-	NULL	2	ajar	belajar→ajar
be-	NULL	2	k*er	bekerja→kerja
per-	NULL	2	NULL	perjelas→jelas
pel-	NULL	2	ajar	pelajar→ajar
pe-	NULL	2	NULL	pekerja→kerja

Tabel 4. 10. Aturan Untuk *Derivational Suffix*

Akhiran	Replacement	Measure Condition	Additional Condition	Contoh
-kan	NULL	2	Prefix bukan anggota {ke, peng}	tarikan→tarik, mengambil→ambil
-an	Null	2	Prefix bukan anggota {di, meng, ter}	makanan→makan, perjanjian→janji
-i	NULL	2	Prefix bukan anggota {ber, ke, peng}	tandai→tanda, mendapati→dapat

Proses stemming menggunakan algoritma porter dapat dilihat pada gambar 4.9.



Gambar 4. 9. Algoritma Porter

4.3. Pembobotan Kata

Setelah melalui tahapan *preprocessing* selanjutnya adalah tahapan pembobotan kata (*Term Weighting*) pada penelitian ini menggunakan algoritma TF-IDF dengan library Scikit Learn seperti pada gambar 4.10.

```
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd

tv = TfidfVectorizer(min_df=0., max_df=1., use_idf=True)
dt_matrix = tv.fit_transform(norm_sentences)
dt_matrix = dt_matrix.toarray()

vocab = tv.get_feature_names()
td_matrix = dt_matrix.T
print(td_matrix.shape)
pd.DataFrame(np.round(td_matrix, 2), index=vocab).head(10)
```

Gambar 4. 10. Proses Pembobotan Kata TF-IDF

Pada gambar 4.10 menjelaskan cara vektorisasi kalimat menggunakan skema TF-IDF yang hasilnya yaitu berupa *term document matrix*. Proses pembobotan kata menggunakan TF-IDF adalah nilai TF (*Term Frequency*) dihitung dengan rumus TF = jumlah frekuensi kata terpilih / jumlah kata dan nilai IDF (*Inverse Document Frequency*) dihitung dengan rumus IDF = log(jumlah dokumen / jumlah frekuensi kata terpilih). Selanjutnya adalah melakukan perkalian antara nilai TF dan IDF untuk mendapatkan hasil akhir seperti contoh pada tabel 4.9 adalah hasil TF_IDF untuk modul pembelajaran dengan judul Modul Sistem Informasi Manajemen.docx dengan jumlah kata 5.137.

Tabel 4. 11. Contoh Hasil TF-IDF

	0	1	2	3	4	5	6	7	8	9	...	77	78	79	80	81	82	83	84
abai	0.0	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.00	...	0.0	0.00	0.00	0.0	0.0	0.0	0.0	0.0
ada	0.0	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.00	...	0.0	0.00	0.16	0.0	0.0	0.0	0.5	0.0
adala	0.0	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.00	...	0.0	0.27	0.00	0.0	0.0	0.0	0.0	0.0
administrasi	0.0	0.0	0.29	0.0	0.0	0.00	0.0	0.0	0.0	0.00	...	0.0	0.00	0.00	0.0	0.0	0.0	0.0	0.0
afta	0.0	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.00	...	0.0	0.00	0.00	0.0	0.0	0.0	0.0	0.0
ahli	0.0	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.00	...	0.0	0.00	0.00	0.0	0.0	0.0	0.0	0.0
aktifitasaktifitas	0.0	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.00	...	0.0	0.00	0.00	0.0	0.0	0.0	0.0	0.0
aktivitas	0.0	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.00	...	0.0	0.00	0.00	0.0	0.0	0.0	0.0	0.0
akuntansi	0.0	0.0	0.00	0.0	0.0	0.25	0.0	0.0	0.0	0.00	...	0.0	0.00	0.00	0.0	0.0	0.0	0.0	0.0
akurat	0.0	0.0	0.00	0.0	0.0	0.24	0.0	0.0	0.0	0.27	...	0.0	0.00	0.00	0.0	0.0	0.0	0.0	0.0

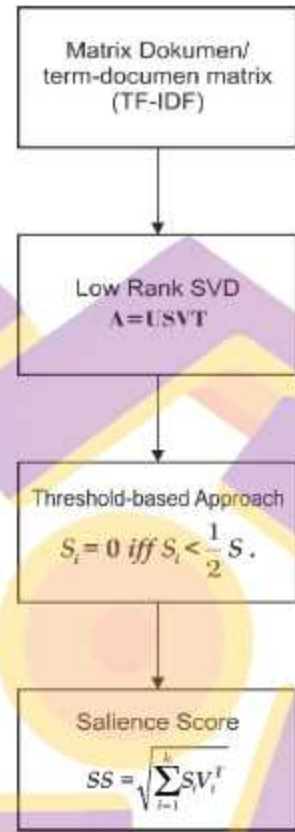
Tabel 4.9 disebut *term-document matrix*. Setiap baris mewakili sebuah kata yang unik, sedangkan setiap kolom mewakili konteks dari mana kata-kata tersebut diambil. Konteks yang dimaksud bisa berupa kalimat, paragraf, atau seluruh bagian dari teks.

4.4. Peringkasan Teks (*Text Summarization*)

Terdapat dua teknik peringkasan teks yaitu peringkasan ekstraktif dan Abstraksi. Peringkasan teks ekstraksi melibatkan identifikasi dan penggalian kalimat yang paling penting dari teks aslinya. Peringkasan ekstraktif lebih mudah diterapkan dan dapat dilakukan dengan cepat menggunakan pendekatan tanpa pengawasan yang tidak memerlukan pelatihan sebelumnya. Sedangkan Abstraksi melibatkan pemahaman gagasan utama teks dan menghasilkan versi ringkasan baru berdasarkan pemahaman itu. Peringkasan abstraktif memiliki keuntungan mampu menghasilkan teks baru, tetapi lebih kompleks untuk diterapkan dan membutuhkan kemampuan generasi bahasa.

Pada penelitian ini menggunakan Teknik peringkasan teks ekstraktif dengan algoritma *Latent Semantic Analysis* (LSA) sehingga model ini tidak membutuhkan data latih untuk menghasilkan peringkasan teks karena algoritma LSA menggunakan metode aljabar yang mengekstrak makna dan kemiripan kalimat dengan informasi tentang kata-kata di kategori tertentu. LSA mengekstrak teks dan mengubahnya menjadi matriks kalimat dan memprosesnya melalui Dekomposisi Nilai Singular (SVD) untuk menemukan kata dan kalimat yang mirip secara semantik. SVD memodelkan hubungan antara kata dan kalimat.

Berikut tahapan secara garis besar pada peringkasan teks otomatis dalam penelitian ini seperti pada gambar 4.11.

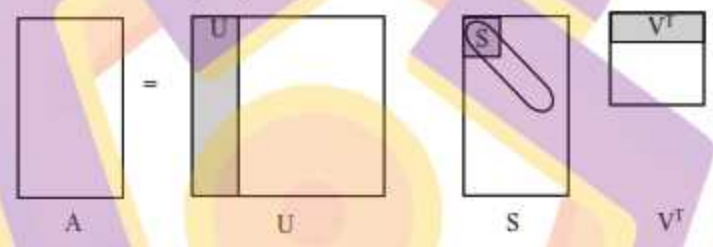


Gambar 4. 11. Tahapan LSA

Pada gambar 4.11 menjelaskan alur proses algoritma LSA dalam peringkasan otomatis modul pembelajaran berbahasa Indonesia yang menggunakan *term-matrix document*. *term-matrix document* didapatkan dari hasil pembobotan kata menggunakan metode TF-IDF yang sebelumnya sudah dilakukan dimana istilah atau kata dalam setiap kalimat dokumen telah diekstraksi untuk membentuk matriks term-dokumen yang memiliki nilai-nilai atau bobot pada setiap katanya

yang merepresentasikan kemunculan kata tersebut dalam dokumen teks yang selanjutnya dalam penelitian ini disebut matriks (matrix).

Tahapan selanjutnya masuk dalam ruang lingkup algoritma SVD, pada penelitian ini menggunakan library SciPy untuk mencari nilai *low-rank* SVD. Pencarian nilai *low-rank* untuk memulihkan *matrix* "asli" (yang sebelumnya terdapat banyak *noise*) agar mendapatkan pendekatan matriks yang paling konsisten sehingga dapat digunakan sebagai pendekatan matrik ideal. Berikut gambaran Teknik low-rank SVD seperti pada gambar 4.12.



Gambar 4. 12. *Low rank approximation* dengan SVD

Berdasarkan gambar 4.12 didapatkan persamaan *low-rank* dengan pendekatan SVD yaitu $A=USV^T$, dimana U adalah matriks orthogonal, S matriks diagonal dan V^T Transpose dari matriks orthogonal.

```
from scipy.sparse.linalg import svds

def low_rank_svd(matrix, singular_count=2):
    u, s, vt = svds(matrix, k=singular_count)
    return u, s, vt
```

```
num_topics = 3

u, s, vt = low_rank_svd(td_matrix, singular_count=num_topics)
print(u.shape, s.shape, vt.shape)
term_topic_mat, singular_values, topic_document_mat = u, s, vt
```

Gambar 4. 13. *Syntax low rank* SVD

Gambar 4.13 adalam implementasi dari pendekatan Teknik *low rank* dengan metode SVD, dimana jumlah topik ditentukan sebanyak 3 topik dalam dokumen teks. Sebagai contoh hasil tahapan *low-rank* SVD pada Modul Sistem Informasi Manajemen.doc seperti gambar berikut.

```
print(td_matrix.shape)

(365, 87)
```

Gambar 4. 14. Bentuk matrix asli

```
print(u.shape, s.shape, vt.shape)

(365, 3) (3,) (3, 87)
```

Gambar 4. 15. Bentuk low-rank svd

Gambar 4.14 adalah bentuk *matrix* dari hasil TF-IDF dan setelah melalui tahapan *low rank* SVD menjadi seperti pada gambar 4.15. setelah melalui tahapan low-rank selanjutnya adalah tahapan *Threshold-based Approach* atau pendekatan berbasis ambang batas.

Penerapan *Threshold-based Approach* untuk menghapus nilai singular dalam penelitian ini menentukan batas penghapusan nilai singular adalah 0,5 atau setengah dari nilai singular yang terbesar jika ada dengan mengkalikan setiap kolom

kalimat istilah dari V kuadrat dengan nilai singular yang sesuai dari S kuadrat, untuk mendapatkan bobot kalimat per topik. Dimana persamaan pendekatan ambang batas seperti pada gambar 4.16 yang di implementasikan dalam syntax seperti pada gambar 4.17.

$$S_i = 0 \; iff \; S_i < \frac{1}{2} S \; .$$

Gambar 4. 16. *Threshold-based Approach*

```
term_topic_mat, singular_values, topic_document_mat = u, s, vt
# remove singular values below threshold
sv_threshold = 0.5
min_sigma_value = max(singular_values) * sv_threshold
singular_values[singular_values < min_sigma_value] = 0
```

Gambar 4. 17. *Syntax Threshold-based Approach*

Setelah melalui tahapan penghapusan nilai kalimat yang terdapat pada dokumen teks di bawah ambang batas yang telah di tentukan maka selanjutnya adalah tahapan *salience score*. Proses *salience score* adalah proses pemberian skor pada kalimat-kalimat yang dianggap penting dengan menghitung jumlah bobot kalimat di seluruh topik dan mengambil akar kuadrat dari skor akhir untuk mendapatkan skor arti-penting (*salience*) untuk setiap kalimat dalam dokumen. Persamaan yang digunakan seperti pada gambar 4.18 yang di implementasikan dalam syntax seperti pada gambar 4.19.

$$SS = \sqrt{\sum_{i=1}^{k} S_i V_i^T}$$

Gambar 4. 18. Persamaan *Salience Score*

```
salience_scores = np.sqrt(np.dot(np.square(singular_values),
                                 np.square(topic_document_mat)))
```

```
num_sentences = 8
top_sentence_indices = (-salience_scores).argsort()[:num_sentences]
top_sentence_indices.sort()
top_sentence_indices
```

```
array([ 8, 15, 16, 32, 49, 50, 52, 53])
```

Gambar 4. 19. *Syntax Salience Score*

Gambar 4.19 adalah implementasi dari persamaan Salience Score dengan penentuan jumlah kalimat tertinggi yang di ambil adalah 8 kalimat. Hasil dari *salience score* yaitu berupa *array* dengan nilai 8 tertinggi dalam tampilan ini menggunakan Modul Sistem Informasi Manajemen.docx sebagai contoh. Tampilan array yang di hasilkan adalah [8, 15, 16, 32, 49, 50, 52, 53]. Nilai dalam array tersebut merepresentasikan 8 kalimat yang dipilih sebagai hasil dari peringkasan teks otomatis

Hasil peringkasan teks otomatis menggunakan algoritma *Latent Semantic Analysis* (LSA) yang berupa paragraph seperti pada contoh Tabel 4.10 berasal dari Modul Pembelajaran Sistem Informasi Manajemen.docx pada link ini: https://bit.ly/simutm yang berjumlah 13.143 kata.

Tabel 4. 12. Hasil Peringkasan Teks

Peringkasan Teks LSA dengan Stemming Nazief-Adriani	Peringkasan Teks LSA dengan Stemming Porter
Informasi adalah data yang diolah menjadi bentuk yang berguna bagi para pemakainya. Sebelum komputer ada, sistem informasi sudah menjadi kebutuhan organisasi. Ini berarti sistem informasi tidak selamanya berbasis komputer. Sebagai tindak lanjut dari tugas manajer tersebut, maka perlu adanya usaha penataan sumberdaya (Manajemen Sumberdaya) termasuk didalamnya manajemen informasi, yakni: ·Sumberdaya harus disusun sedemikian rupa sehingga setiap saat diperlukan dapat segera dimanfaatkan - perlu dilakukan modifikasi ·Sumberdaya harus dimanfaatkan semaksimal mungkin ·Sumberdaya harus selalu diperbaharui. Manajer memastikan bahwa data mentah yang diperlukan terkumpul dan kemudian diproses menjadi informasi yang berguna. Peningkatan kemampuan komputer, Manajemen Data dan Komunikasi : Trend Manajemen Data Ditinjau dari Segi Teknik Manajemen ·File management dan organization hanya untuk satu aplikasi tertentu untuk beberapa aplikasi untuk corporate data files (diperlukan database sistems) perlu dibuat data dictionary, bukan hanya sekedar data definitions. Ditinjau dari Segi Pengelolaan Data ·Terjadi pergeseran model pengolahan data, yang tadinya dilakukan secara tersentralisasi (terpusat) kini menjadi pengolahan data terdesentralisasi atau pengolahan terdistribusi. Ditinjau dari Segi Asal Data ·Berdasarkan asal data yang akan diolah, yang kebanyakan berasala dari Data Internal kini bergeser dengan melibatkan Data Eksternal. Ditinjau dari Segi Jenis Data ·Pengolahan data dilakukan berdasarkan data yang dikumpulkan sehingga menghasilkan informasi.	Sebelum komputer ada, sistem informasi sudah menjadi kebutuhan organisasi. Sebagai tindak lanjut dari tugas manajer tersebut, maka perlu adanya usaha penataan sumberdaya (Manajemen Sumberdaya) termasuk didalamnya manajemen informasi, yakni: Sumberdaya harus disusun sedemikian rupa sehingga setiap saat diperlukan dapat segera dimanfaatkan - perlu dilakukan modifikasi · Sumberdaya harus dimanfaatkan semaksimal mungkin · Sumberdaya harus selalu diperbaharui : Manajer memastikan bahwa data mentah yang diperlukan terkumpul dan kemudian diproses menjadi informasi yang berguna. Peningkatan kemampuan komputer, Manajemen Data dan Komunikasi : Trend Manajemen Data Ditinjau dari Segi Teknik Manajemen · File management dan organization hanya untuk satu aplikasi tertentu untuk beberapa aplikasi untuk corporate data files (diperlukan database sistems) perlu dibuat data dictionary, bukan hanya sekedar data definitions. Ditinjau dari Segi Pengelolaan Data ·Terjadi pergeseran model pengolahan data, yang tadinya dilakukan secara tersentralisasi (terpusat) kini menjadi pengolahan data terdesentralisasi atau pengolahan terdistribusi. Ditinjau dari Segi Asal Data · Berdasarkan asal data yang akan diolah, yang kebanyakan berasala dari Data Internal kini bergeser dengan melibatkan Data Eksternal. Ditinjau dari Segi Jenis Data · Pengolahan data dilakukan berdasarkan data yang dikumpulkan sehingga menghasilkan informasi. Dengan perkataan lain, yang dulunya hanya : melakukan pertukaran data antar organisasi atau unit organisasi, terus meningkat menjadi pertukaran informasi (yang merupakan hasil pengolahan dari data). Manajer juga dijumpai dalam bidang fungsional perusahaan, tempat berbagai sumberdaya dipisahkan menurut jenis pekerjaan yang dilakukan.

4.5. Evaluasi

Evaluasi ini dilakukan untuk mengukur performa dari model yang di gunakan. Pada penelitian ini akan menggunakan metode evaluasi ROUGE.

ROUGE merupakan standar pengujian peringkasan teks berbahasa indonesia untuk mengukur kualitas hasil ringkasan dengan membandingkan hasil peringkasan yang dilakukan oleh sistem dan hasil ringkasan oleh manusia(manual) dengan mencari nilai dari precision, recall, dan f-measure. Penelitian ini, akan menggunakan ROUGE-N (ROUGE 1 dan ROUGE 2) dan ROUGE-L yang terdiri atas precision, recall, dan f-measure sebagai scoring untuk mengukur kinerja sistem, mengikuti banyak penelitian di bidang peringkasan dokumen seperti pada penelitian dengan judul "Peringkasa teks otomatis pada artikel berbahasa indonesia menggunakan algoritma Lexrank" (Halimah et al., 2022).

Persamaan pada perhitungan ROUGE:

$$ROUGE - 1\,recall = \frac{jumlah\ unigram\ kata\ yang\ sama}{Total\ kata\ di\ ringkasan\ manual}$$

$$ROUGE - 1\,precision = \frac{jumlah\ unigram\ kata\ yang\ sama}{Total\ kata\ di\ ringkasan\ sistem}$$

$$ROUGE - 2\,recall = \frac{jumlah\ bigram\ kata\ yang\ sama}{Total\ kata\ di\ ringkasan\ manual}$$

$$ROUGE - 2\,precision = \frac{jumlah\ bigram\ kata\ yang\ sama}{Total\ kata\ di\ ringkasan\ sistem}$$

$$ROUGE - L\,recall = \frac{LCS\ (sistem, manual)}{Total\ kata\ di\ ringkasan\ manual}$$

$$ROUGE - L\,precision = \frac{LCS\ (sistem, manual)}{Total\ kata\ di\ ringkasan\ sistem}$$

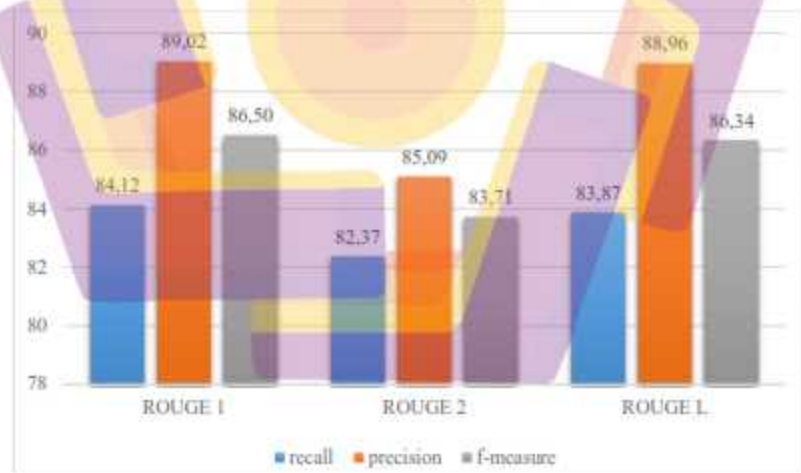
$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Pada penelitian ini perhitungan rouge menggunakan library python rouge dan menampilkan hasil perhitungan rouge dengan nilai dari 0 sampai 1 seperti pada gambar 4.20 yang selanjutnya selanjtnya dalam penelitian ini di tampilkan dalam bentuk persen (%).

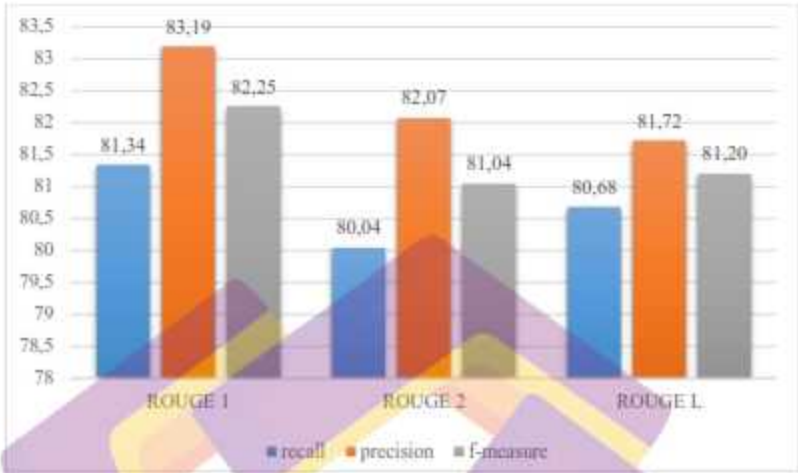
```
!pip install rouge
```

```
from rouge import Rouge
rouge = Rouge()
rouge.get_scores(pakar, abc, avg=True, ignore_empty=True)
```

Gambar 4. 20. Perhitungan *Rouge*

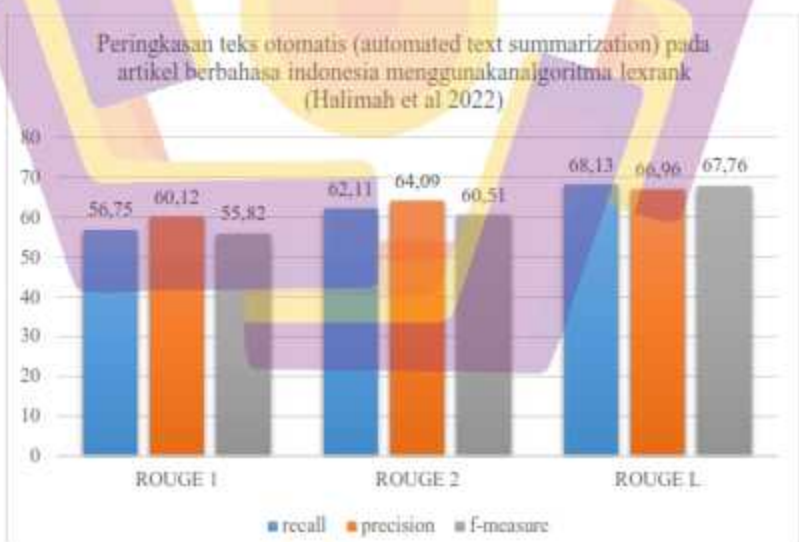


Gambar 4. 21. Grafik hasil peringkasan teks stemming Nazief-Adriani

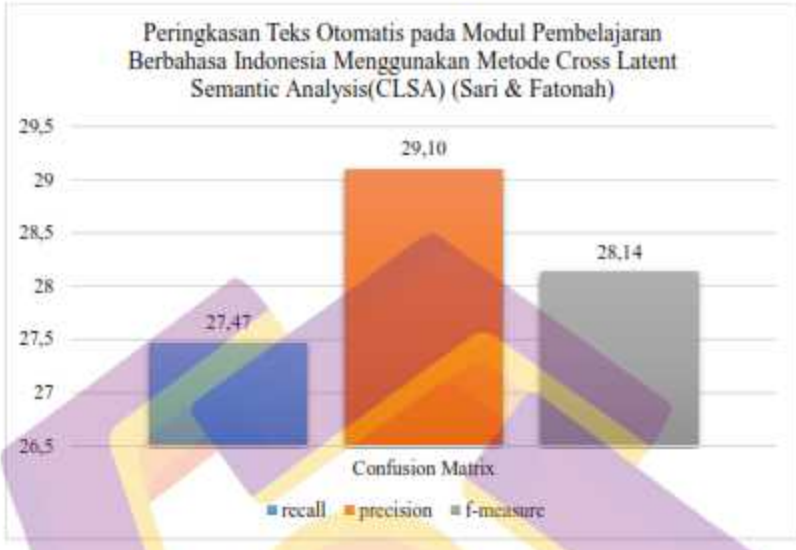


Gambar 4. 22. Grafik hasil peringkasan teks stemming Porter

Berikut beberapa grafik hasil peringkasan teks dari penelitian terdahulu:



Gambar 4. 23. Grafik hasil peringkasan teks dengan algoritma Lexrank



Gambar 4. 24. Grafik hasil peringkasan teks dengan algoritma CLSA

Pada Tabel 4.11 berikut ini adalah tabel Perbandingan nilai rata-rata seluruh pengujian hasil perhitungan *Recall, Precision* dan *F-Measure* dengan penelitian sebelumnya:

Tabel 4. 13. Perbandingan Hasil pengujian dalam persen (%)

	Precision	***Recall***	***F-Measure***
Halimah et al	63,72	62,33	61,36
Yunita Maulida & Nenden Fatonah	29,10	27,47	28,14
Peringkasan Teks Penelitian ini	85,01	82,03	83,43

Selain dilakukan pengujian menggunakan metode ROUGE-N dan ROUGE-L juga dilakukan perhitungan kecepatan proses peringkasan teks otomatis pada kedua algoritma stemming dengan hasil seperti pada tabel 4.12, dimana seluruh data

modul pembelajaran dapat dilihat pada link ini: https://bit.ly/modulpembelajaranutm .

Tabel 4. 14. Kecepatan Proses Algoritma Stemming

No	Judul Modul	Jumlah Kata	Nazief & Adriani	Porter
			Durasi (detik)	Durasi (detik)
1	Modul Sistem Operasi.docx	8.831	16	7
2	Modul Technopreneurship 1.docx	8.952	12	2
3	Modul Kecakapan Antarpersonal.pdf	28.905	45	12
4	Modul Technopreneurship II.pdf	7.213	13	2
5	Modul Manajemen operasi perkantoran.docx	6.765	8	1
6	Modul Efisiensi Kantor.docx	6.262	6	2
7	Modul Entrepreneurship 2.docx	7.213	13	2
8	Modul KNB Indonesia.docx	7.062	7	2
9	Modul Komunikasi Niaga.docx	5.991	5	3
10	Modul Manajemen Kearsipan.docx	11.355	16	2
11	Modul Manajemen Organisasi dan Produksi.docx	8.492	8	1
12	Modul Pengantar Ilmu Administrasi.docx	6.195	6	2
13	Modul Sistem Infromasi Manajemen.docx	13.143	19	2
14	Modul Statistik Deskriptif.docx	7.154	8	2
15	Modul Analisis Laporan Keuangan.pdf	16.369	24	6
16	Modul Manajemen Perkantoran 2020.pdf	23.003	53	15
17	Modul Metodologi Penelitian Bisnis.Pdf	27.311	31	7
18	Modul Penganggaran Perusahaan.pdf	17.992	21	11
19	Modul Statistik 1.pdf	16.064	23	7
20	Modul Statistik 2.pdf	28.679	27	10
21	Modul Studi Kelayakan Bisnis.pdf	27.709	22	9
22	Modul Surat - Menyurat.pdf	15.009	21	9
23	Modul UMKM dan Koperasi.pdf	18.649	26	6
24	Modul-Perpajakan.pdf	18.674	29	7
25	Manajemen Organisai Perkantoran.pdf	15.725	22	10
Rata-Rata		**14.349**	**19**	**6**

BAB V

PENUTUP

5.1. Kesimpulan

Terdapat beberapa kesimpulan yang dapat disimpulkan pada penelitian ini.

1. Proses stemming dokumen teks berbahasa Indonesia menggunakan Algoritma Porter membutuhkan waktu yang lebih singkat yaitu dengan rata-rata 6 detik, dibandingkan *stemming* menggunakan Algoritma Nazief & Adriani yaitu rata-rata proses 19 detik yang artinya selisih 13 detik akan tetapi hasil akurasi peringkasan teks otomatis menggunakan Algoritma Nazief & Adriani mendapatkan nilai lebih tinggi dengan nilai *Precision, Recall* dan F-*Measure* yaitu 87,69%, 83,41%, 85,37% di bandingkan Algoritma Porter yaitu 82,33%, 80,65%, 81,50%.
2. Percobaan algoritma *Latent Semantic Analysis (LSA)* dalam peringkasan dokumen modul pembelajaran berbahasa indonesia di Universitas Teknologi Mataram memiliki tingkat akurasi rata-rata 83,49% dan dengan tingkat akurasi tersebut hasil peringkasan teks otomatis pada modul pembelajaran ini lebih baik di bandingkan dengan penelitian sebelumnya yang memiliki tingkat akurasi di bawah 70%.

5.2. Saran

Dari penelitian yang dilakukan, terdapat saran untuk penelitian yang akan datang. Adapun sarannya yaitu dengan memperhatikan *library* python yang digunakan untuk merubah file dokumen berformat docx atau pdf ke dalam bentuk

teks atau paragraph untuk di olah karena pada umumnya *library* python digunakan untuk dokumen berbahasa inggris sehingga jika di terapkan pada file berbahasa Indonesia hasil ekstraksi dokumen dalam penyusunan kalimat sangat tidak tepat contohnya kata "ingin" akan di ekstrak menjadi "i ngin" dan hal tersebut akan mempengaruhi hasil peringkasan teks menjadi rendah. Untuk penelitian selanjutnya bisa menggunakan *library* ekstraksi file docx dan pdf yang mendukung Bahasa Indonesia.

DAFTAR PUSTAKA

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Al-Hafiidh Firman, D., Fahrur Rozi, I., & Kusumaning Putri, I. (2022). Peringkasan teks otomatis pada portal berita olahraga menggunakan metode maximum marginal relevance. *JIP (Jurnal Informatika Polinema)*, *8*(3), 21–30.

Ayu Syahfitri, R., Kurniawan, A., & Irsan Humaidy, M. (2022). Penerapan Algoritma Maximum Marginal Relevance Dalam Peringkasan Teks Secara Otomatis. *Bulletin of Data Science*, *1*(2), 49–56. https://ejurnal.seminar-id.com/index.php/bulletinds

Gupta, H., & Patel, M. (2021). Method of Text Summarization Using Lsa and Sentence Based Topic Modelling with Bert. *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*, 511–517. https://doi.org/10.1109/ICAIS50930.2021.9395976

Halimah, Agustian, S., & Ramadhani, S. (2022). Peringkasan teks otomatis (automated text summarization) pada artikel berbahasa indonesia menggunakan algoritma lexrank. *Jurnal CoSciTech (Computer Science and Information Technology)*, *3*(3), 371–381. https://doi.org/10.37859/coscitech.v3i3.4300

Husniah, F., Agustian, S., & Afrianty, I. (2022). *peringkasan teks otomatis artikel berbahasa indonesia menggunakan algoritma textrank*.

Jayadianti Herlina, Damayanti Ruth, & Juwairiah. (2020). Latent Semantic Analysis (Lsa) Danutomatic Text Summarization (Ats) Dalam Optimasi Pencarian Artikel Covid. *Seminar Nasional Informatika*.

Kumar, A., Sharma, A., & Nayyar, A. (2020). Fuzzy Logic based Hybrid Model for Automatic Extractive Text Summarization. *ACM International Conference Proceeding Series*, 7–15. https://doi.org/10.1145/3385209.3385235

Malak Diana D, Bagas Satya D, & Faisal Rahutomo. (2020). Penerapan algoritma score-based pada peringkasan teks cerpen otomatis. *Seminar informatika aplikatif polinema*, 2020.

Putu Satwika, I., & Syahk Alam, H. (2020). *algoritma stemming dalam bahasa bali menggunakan pendekatan n-gram* (vol. 2, issue 2).

Rahim Arham. (2022). *evaluasi esai otomatis dengan algoritma nazief & adriani dan winnowing* (Vol. 4, Issue 1).

Rani Balmuri, K., Rao Mangu, V., Konda, S., & Gunda, M. (2021). *Telugu Text Summerization Using Lstm Deep Learning*. https://www.researchgate.net/publication/354623414

Rozi, I. F., Suarjuna Batubulan, K., Rusbandi, M., Informatika, T., Informasi, T., & Malang, P. N. (2021). Otomatisasi Peringkasan Teks Pada Dokumen Hukum Menggunakan Metode Latent Semantic Analysis. *JIP (Jurnal Informatika Polinema)* , *7*(3), 9–15.

Simanjuntak, I. Z. (2022). *Informasi dan Teknologi Ilmiah (INTI)*.

Yunita Maulidia Sari, & Nenden Siti Fatonah. (2021). Peringkasan Teks Otomatis pada Modul Pembelajaran Berbahasa Indonesia Menggunakan Metode Cross Latent Semantic Analysis (CLSA). *Jurnal Edukasi Dan Penelitian Informatika*. www.kompas.com.

Zezen, A., Abidin, Z., & Nurjanah, E. (2020). Sistem Peringkas Teks Otomatis Multi Dokumen Kliping Artikel Berita Gempa Menggunakan Metode Tf-Idf. In *STMIK Subang* (Vol. 13, Issue 1).