Jurnal Kecerdasan Buatan dan Teknologi Informasi
Vol. 4, No. 1, January 2025, hlm. 53-59
e-ISSN: 2964-2922, p-ISSN: 2963-6191
DOI : https://doi.org/10.69916/jkbti.v4i1.202

PERBANDINGAN ALGORITMA *NAZIEF-ADRIANI* DAN *PORTER* UNTUK PERINGKASAN TEKS OTOMATIS DENGAN *LATENT SEMANTIC ANALYSIS* PADA MODUL PEMBELAJARAN BERBAHASA INDONESIA

St Tuhpatussania[1], Surni Erniwati[*2]

[1]Teknik Informatika, Fakultas Teknologi dan Informasi Komputer, Universitas Teknologi Mataram, Mataram, Indonesia
[2]Manajemen Informatika, Fakultas Vokasi, Universitas Teknologi Mataram, Mataram, Indonesia
Email: [1]asna.tuhfah@gmail.com, [2]mentari1990@gmail.com

(Diterima : 4 Desember 2024, Direvisi : 27 Desember 2024, Disetujui : 4 Januari 2025)

Abstrak

Penelitian ini mengembangkan sistem peringkasan teks otomatis untuk modul pembelajaran berbahasa Indonesia menggunakan metode *Latent Semantic Analysis*. Tantangan utama peringkasan teks dalam Bahasa Indonesia mencakup kompleksitas struktur bahasa dan penggunaan imbuhan, yang memerlukan proses prapemrosesan teks secara menyeluruh, termasuk stemming. Penelitian ini membandingkan dua algoritma stemming, yaitu *Nazief-Adriani* dan *Porter*, untuk mengubah kata berimbuhan menjadi bentuk dasar. Metode ini diawali dengan pemecahan kalimat, pembersihan teks, penghapusan kata tidak penting, dan pembobotan kata menggunakan *Term Frequency-Inverse Document Frequency*. Tahap selanjutnya adalah analisis hubungan semantik antar kata dan kalimat menggunakan *Singular Value Decomposition* untuk menghasilkan matriks term-dokumen yang diproses menjadi salience score. Hasil evaluasi menunjukkan bahwa algoritma *Nazief-Adriani* memiliki akurasi lebih tinggi dengan nilai *precision*, *recall*, dan *F-measure* masing-masing sebesar 87,69%, 83,41%, dan 85,37%, dibandingkan *Porter* yang hanya mencapai rata-rata 81,50%. Algoritma *Latent Semantic Analysis* memberikan tingkat akurasi rata-rata sebesar 83,49%, lebih unggul dibandingkan penelitian sebelumnya. Kesimpulan penelitian ini menegaskan efektivitas metode *Latent Semantic Analysis* untuk peringkasan teks otomatis dan superioritas algoritma *Nazief-Adriani* dalam menghasilkan akurasi yang lebih baik dan merekomendasikan pengembangan sistem yang lebih efisien dan mendukung pemrosesan Bahasa Indonesia secara optimal.

Kata kunci: LSA, modul pembelajaran, peringkasan, *stemming*, teks indonesia.

AUTOMATIC TEXT SUMMARIZATION IN INDONESIAN LANGUAGE LEARNING MODULES USING THE LATENT SEMANTIC ANALYSIS APPROACH

Abstract

This study develops an automatic text summarization system for Indonesian language learning modules using the Latent Semantic Analysis (LSA) method. The primary challenges in text summarization for Bahasa Indonesia include the complexity of the language structure and the use of affixes, necessitating comprehensive text preprocessing, including stemming. The research compares two stemming algorithms: Nazief-Adriani and Porter, to transform affixed words into their base forms. The methodology begins with sentence segmentation, text cleaning, removal of non-essential words, and word weighting using Term Frequency-Inverse Document Frequency. The next step involves analyzing semantic relationships between words and sentences using Singular Value Decomposition to produce a term-document matrix, which is then processed into salience scores. Evaluation results indicate that the Nazief-Adriani algorithm achieves higher accuracy, with precision, recall, and F-measure values of 87.69%, 83.41%, and 85.37%, respectively, compared to Porter, which averages 81.50%. The LSA algorithm provides an average accuracy of 83.49%, surpassing previous studies. The conclusion of this research underscores the effectiveness of the Latent Semantic Analysis method for automatic text summarization and the superiority of the Nazief-Adriani algorithm in achieving better accuracy, recommending the development of more efficient systems to optimally support Indonesian language processing.

Keywords: *indonesian text, learning module, LSA, stemming, summarization.*

1. PENDAHULUAN

Kemajuan teknologi informasi telah mengubah cara manusia memperoleh dan mengolah informasi. Masyarakat saat ini dapat dengan mudah mengakses informasi melalui berbagai media, baik itu internet, buku, artikel, majalah, maupun koran [1]. Di era digital ini, kecepatan dan kemudahan akses terhadap informasi menjadi prioritas menyeluruh. Dalam konteks inilah, teknologi peringkasan teks otomatis menjadi sangat relevan, karena membantu pengguna memperoleh inti informasi dari dokumen panjang dalam waktu yang lebih singkat. Peringkasan teks otomatis bertujuan untuk menyajikan esensi dari sebuah dokumen tanpa memerlukan penyuntingan manusia. Dengan peringkasan teks otomatis, ide pokok atau informasi penting dapat diperoleh secara ringkas dan tepat, tanpa mengubah makna asli dari dokumen tersebut [1]. Terdapat dua pendekatan utama dalam peringkasan teks, yaitu peringkasan ekstraktif dan abstraktif. Peringkasan ekstraktif memilih kalimat-kalimat dari dokumen asli dan menggabungkannya menjadi ringkasan, tanpa mengubah struktur kalimat [2]. Sedangkan peringkasan abstraktif lebih kompleks, karena melibatkan penyusunan ulang kalimat berdasarkan kata-kata inti dalam dokumen asli [3]. Peringkasan abstraktif lebih sulit dilakukan karena membutuhkan pemahaman yang mendalam terhadap konteks dokumen, sementara peringkasan ekstraktif lebih sederhana namun bisa menghasilkan ringkasan yang kurang kohesif [4][5].

Peringkasan teks otomatis dalam bahasa Indonesia memiliki tantangan tambahan dibandingkan dengan bahasa Inggris. Bahasa Indonesia kaya akan imbuhan seperti awalan, akhiran, konfix, dan infix, yang dapat mempersulit proses peringkasan [6]. Untuk menghasilkan ringkasan yang akurat, kata-kata dalam dokumen perlu diubah menjadi kata dasar (*root word*) melalui proses stemming [7]. Dalam bahasa Indonesia, terdapat kurang lebih 35 imbuhan yang diakui dalam Kamus Besar Bahasa Indonesia (KBBI), yang harus dihapus selama proses stemming untuk mencegah redundansi kata [6]. Algoritma stemming seperti *Nazief-Adriani* dan *Porter* sering digunakan dalam penelitian yang berhubungan dengan teks bahasa Indonesia [8], [9].

Penelitian terdahulu yang membahas peringkasan teks otomatis umumnya menggunakan berbagai metode. Syahfitri et al. (2022) menggunakan algoritma Maximum Marginal Relevance, yang mengandalkan pembobotan kalimat berdasarkan urutan bobotnya. Namun, hasil ringkasan yang diperoleh tidak memiliki urutan sistematis yang baik, sehingga kurang optimal dalam memberikan gambaran menyeluruh dari dokumen [10]. Penelitian lainnya oleh Sari dan Nenden (2021) menggunakan metode Cross *Latent Semantic Analysis (CSLA)* untuk peringkasan modul pembelajaran berbahasa Indonesia, namun akurasi yang dihasilkan masih rendah, terutama pada compression rate sebesar 20%, dengan f-measure hanya mencapai 0.3853 [11].

Algoritma LSA telah terbukti efektif dalam peringkasan teks, terutama untuk bahasa Indonesia. Algoritma ini menggunakan teknik *Singular Value Decomposition (SVD)* untuk mengidentifikasi hubungan semantik antar kata dan kalimat dalam dokumen [12]. Beberapa penelitian telah menunjukkan bahwa LSA dapat menghasilkan ringkasan yang mendekati hasil ringkasan manual oleh manusia, meskipun pada compression rate yang berbeda. Rozi et al. (2021) menguji LSA pada dokumen hukum berbahasa Indonesia dan mendapatkan akurasi sebesar 75% pada compression rate 50% [12]. Meskipun demikian, algoritma LSA masih memerlukan peningkatan dalam hal akurasi dan efisiensi, terutama dalam konteks bahasa Indonesia yang kompleks.

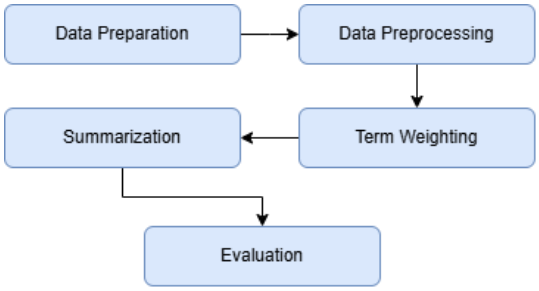
Berdasarkan kajian literatur, metode stemming memiliki peran penting dalam meningkatkan kinerja peringkasan teks otomatis. Algoritma *Nazief-Adriani*, yang berbasis pada kamus bahasa Indonesia, dan algoritma *Porter*, yang lebih sederhana dengan menghapus imbuhan, sering digunakan untuk memperbaiki proses stemming dalam teks bahasa Indonesia. Penelitian oleh Pamungkas et al. (2023) menunjukkan bahwa algoritma *Nazief-Adriani* memiliki akurasi yang lebih tinggi dibandingkan dengan Tala, dengan akurasi rata-rata sebesar 78.47% [13]. Namun, waktu pemrosesan yang dibutuhkan oleh *Nazief-Adriani* lebih lama dibandingkan dengan *Porter*, sehingga perlu adanya kompromi antara kecepatan dan akurasi dalam penerapannya.

Kebutuhan untuk mengembangkan metode peringkasan teks otomatis yang lebih efektif dan efisien dalam konteks bahasa Indonesia tentunya sangat diperlukan. Penelitian sebelumnya menunjukkan bahwa akurasi peringkasan teks otomatis dalam bahasa Indonesia masih berada di bawah harapan, terutama pada compression rate yang rendah. Oleh karena itu, penelitian ini bertujuan untuk melakukan peringkasan teks otomatis pada modul pembelajaran berbahasa Indonesia menggunakan metode *Latent Semantic Analysis* (LSA), serta membandingkan dua algoritma stemming, yaitu *Nazief-Adriani* dan *Porter*, dalam menentukan kata dasar. Dengan demikian, diharapkan hasil penelitian ini dapat memberikan kontribusi signifikan dalam pengembangan teknologi peringkasan teks otomatis, terutama dalam konteks modul pembelajaran berbahasa Indonesia.

2. METODE PENELITIAN

Penelitian ini merupakan penelitian deskriptif kuantitatif. Penelitian deskriptif bertujuan untuk menggambarkan atau menjelaskan fakta secara sistematis, faktual, dan akurat tanpa menguji hipotesis atau hubungan antar variabel. Metodologi kuantitatif digunakan untuk mendapatkan hasil yang dapat digeneralisasikan, dengan fokus pada keluasan data dibandingkan kedalaman analisis.

Berikut tahapan-tahapan Metodologi penelitian seperti pada Gambar 1.



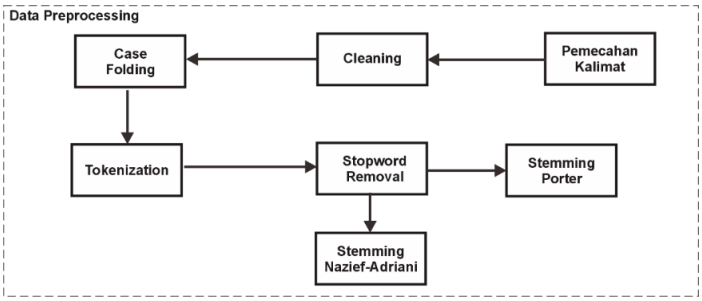
Gambar 1. Metodologi Penelitian Peringkasan Teks

2.1 *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.

2.2 *Data Preprocessing*

Tahapan *preprocessing* dalam penelitian ini mencakup 6 proses yaitu dimulai dengan pemecahan kalimat lalu *cleaning*, *case folding*, *tokenization*, *stopword removal* dan yang terakhir stemming dengan urutan tahapan seperti pada Gambar 2.



Gambar 2. Tahapan *Preprocessing*

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 pemecahan kalimat Pada tahapan pemecahan kalimat yang memecah dokumen yang telah diinputkan menjadi per kalimat dengan delimeter tertentu, selanjutnya kumpulan kalimat yang telah dipecah melalui 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. 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. Setelah tahapan case folding yaitu tokenization Dimana tahapan ini berfungsi untuk memotong atau memisahkan atau memecah kalimat menjadi perkata berdasarkan spasi sebagai pemotong/ pemisah/ pemecah kata tersebut. Dalam tahap *stopword removal* 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. Tahapan yang terakhir pada *preprocessing* adalah Stemming yaitu tahapan yang bertujuan untuk mengubah kata berimbuhan menjadi kata dasar dimana dalam penelitian ini akan menggunakan dua metode stemming yaitu stemming *Porter* dan Stemming *Nazief-Adriani*. 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.

2.3 *Term Weighting*

Term Frequency-Inverse Document Frequency (TF-IDF): merupakan ekstraksi kalimat dengan cara memberikan nilai atau bobot pada kata [14]. 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*) [15].

2.4 *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 [16].

2.5 *Evaluation*

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 system.

3. HASIL DAN PEMBAHASAN

Terdapat 25 modul pembelajaran yang digunakan dengan ekstensi file pdf dan docx yang tidak terkunci dengan rata-rata jumlah kata yang terkandung pada masing-masing modul adalah 14.349 kata. Masing-masing modul akan melalui tahap *preprocessing* data. Pada tahapan pemecahan kalimat, cleaning, case folding, tokenization, dan stopword removal menggunakan library python Aspose.Words, nltk.sent_tokenize, nltk.word_tokenize dan library nltk.corpus. tahapan terakhir *preprocessing* adalah stemming yang menggunakan dua metode stemming untuk menemukan kata dasar yaitu metode stemming *Porter* dan Metode Stemming *Nazief-Adriani*. Pada Metode Stemming *Nazief-Adriani* Tahapan akan membuang semua imbuhan yang terdapat pada suatu kata dengan menggunakan aturan dan berdasarkan kamus. Metode ini menggunakan library Sastrawi yang mengimplementasikan algortima Nazief Adriani sedangkan untuk Metode *Porter* akan menghapus imbuhan tanpa berdasarkan kamus dan dengan menggunakan aturan-aturan yang sudah ditentukan.

Tabel 1. Hasil Stemming

Isi Paragraf Asli	**Hasil *preprocessing***
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

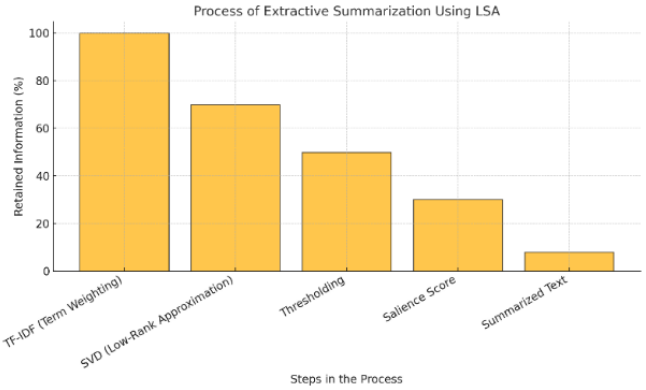
Pada Table 1 adalah contoh kalimat asli dan kalimat yang sudah melalui tahap *preprocessing*. Susunan kata yang sudah melalui tahap *preprocessing* akan berisi kata-kata dasar tanpa tanda baca. Kata-kata tersebut akan di hitung bobotnya menggunakan TF-IDF.

Tabel 2. Hasil TF-IDF

Term	**Bobot**
akuntansi	0.188982
akurat	0.188982
bantu	0.188982
bentuk	0.188982
capai	0.188982
informasi	0.566947
lapor	0.188982
manfaat	0.188982
percaya	0.188982
saji	0.188982
sistem	0.377964
uang	0.188982
unggul	0.188982
usaha	0.377964

Tabel 2 menunjukkan bobot TF-IDF dari setiap kata unik dalam satu dokumen. Kolom pertama menampilkan daftar kata, sementara kolom kedua menunjukkan nilai TF-IDF untuk masing-masing kata. Bobot yang tinggi

menunjukkan kata dengan frekuensi dominan dalam dokumen tetapi jarang muncul di dokumen lain, sedangkan bobot rendah menunjukkan kata yang kurang penting. Karena hanya ada satu dokumen, bobot dihitung berdasarkan distribusi kata dalam dokumen tersebut, dengan kata seperti "sistem" dan "informasi" memiliki bobot lebih tinggi karena kemunculannya yang lebih dominan.



Gambar 3. Proses Peringkasan Teks

Gambar 3 adalah alur proses peringkasan teks menggunakan metode LSA, mulai dari pembobotan kata dengan TF-IDF, reduksi dimensi menggunakan SVD, hingga pemilihan kalimat penting dengan Salience Score. Setiap langkah mengurangi informasi yang tidak relevan, menyisakan delapan kalimat utama sebagai hasil peringkasan.

Tabel 3. 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	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

dikumpulkan sehingga menghasilkan informasi	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.

Tabel 4 Hasil Akurasi dan Durasi Proses

	ROUGE			**Durasi Rata-rata (Detik)**
	Recall	***Precision***	***F-Measure***	
Nazief-Adriani	83.45333	87.69	85.51667	19
Porter	80.68667	82.32667	81.49667	6

Hasil peringkasan seperti pada contoh Tabel 3 selanjutnya akan di evaluasi 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. 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. Pada Tabel 4 menampilkan perbandingan hasil akurasi rata-rata *ROUGE* pada *Recall*, *Precision* dan *F-Measure* serta hasil durasi proses masing-masing metode dalam melalukan peringkasan teks. Dimana dari sisi nilai akurasi *ROUGE*, metode *Nazief-Adriani* mendapat nilai rata-rata lebih tinggi di banding metode *Porter* yang diterapkan saat proses *stemming*.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan metode *Latent Semantic Analysis* (LSA) untuk peringkasan teks otomatis pada modul pembelajaran berbahasa Indonesia. Hasil evaluasi menunjukkan bahwa algoritma *stemming Nazief-Adriani* lebih unggul dibandingkan algoritma *Porter* dalam menghasilkan ringkasan yang akurat. *Nazief-Adriani* mencapai nilai rata-rata *precision* sebesar 87,69%, *recall* sebesar 83,41%, dan *F-measure* sebesar 85,37%, sedangkan *Porter* hanya memperoleh rata-rata precision sebesar 82,33%, recall sebesar 80,69%, dan *F-measure* sebesar 81,50%. Perbedaan ini menunjukkan bahwa *Nazief-Adriani* lebih efektif dalam menangani kompleksitas morfologi Bahasa Indonesia, meskipun membutuhkan waktu pemrosesan rata-rata 19 detik, lebih lama dibandingkan *Porter* yang hanya memerlukan 6 detik.

Algoritma LSA sendiri memberikan akurasi rata-rata sebesar 83,49%, lebih baik dibandingkan beberapa penelitian sebelumnya. Kombinasi *preprocessing* yang melibatkan tokenisasi, stemming, dan pembobotan kata menggunakan TF-IDF berperan penting dalam mendukung akurasi ini. Penelitian ini merekomendasikan pengembangan sistem peringkasan yang lebih efisien, terutama dengan integrasi algoritma yang mampu mengurangi waktu pemrosesan tanpa mengorbankan akurasi.

DAFTAR PUSTAKA

[1] D. F. Al-Hafiidh, I. F. Rozi, and I. K. Putri, "Peringkasan Teks Otomatis Pada Portal Berita Olahraga Menggunakan Metode Maximum Marginal Relevance," *JIP (Jurnal Inform. Polinema)*, vol. 8, no. 3, pp. 21–30, 2022.

[2] Halimah, S. Agustian, and S. Ramadhani, "Peringkasan teks otomatis (automated text summarization) pada artikel berbahasa indonesia menggunakan algoritma lexrank," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 3, no. 3, pp. 371–381, Dec. 2022, doi: 10.37859/coscitech.v3i3.4300.

[3] F. Husniah, S. Agustian, and I. Afrianty, "Peringkasan Teks Otomatis Artikel Berbahasa Indonesia Menggunakan Algoritma Textrank," 2022.

[4] N. Giarelis, C. Mastrokostas, and N. Karacapilidis, "Abstractive vs. Extractive Summarization: An Experimental Review," 2023. doi: 10.3390/app13137620.

[5] A. A. Syed, F. L. Gaol, and T. Matsuo, "A survey of the state-of-the-art models in neural abstractive text summarization," 2021. doi: 10.1109/ACCESS.2021.3052783.

[6] I. Z. Simanjuntak, "Analisa Kombinasi Algoritma Stemming Dan Algoritma Soundex Dalam Pencarian Kata Bahasa Indonesia," *Inf. dan Teknol. Ilm.* , vol. 10, no. 1, pp. 24–30, 2022, Accessed: Dec. 30, 2022. [Online]. Available: http://ejurnal.stmik-budidarma.ac.id/index.php/inti/article/view/5040

[7] I. P. Satwika and H. S. Alam, "Algoritma Stemming Dalam Bahasa Bali Menggunakan Pendekatan N-Gram," 2020.

[8] E. Lindrawati, E. Utami, and A. Yaqin, "Comparison of Modified Nazief&Adriani and Modified Enhanced Confix Stripping algorithms for Madurese Language Stemming," *INTENSIF J. Ilm. Penelit. dan Penerapan Teknol. Sist. Inf.*, vol. 7, no. 2, 2023, doi: 10.29407/intensif.v7i2.20103.

[9] J. Jumadi, D. S. Maylawati, L. D. Pratiwi, and M. A. Ramdhani, "Comparison of *Nazief-Adriani* and Paice-Husk algorithm for Indonesian text stemming process," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1098, no. 3, 2021, doi: 10.1088/1757-899x/1098/3/032044.

[10] A. Kurniawan and M. I. Humaidy, "Penerapan Algoritma Maximum Marginal Relevance Dalam Peringkasan Teks Secara Otomatis," *Bull. Data Sci.*, vol. 1, no. 2, pp. 49–56, 2022, [Online]. Available: https://ejurnal.seminar-id.com/index.php/bulletinds

[11] Y. M. Sari and N. S. Fatonah, "Peringkasan Teks Otomatis pada Modul Pembelajaran Berbahasa Indonesia Menggunakan Metode Cross *Latent Semantic Analysis* (CLSA)," *J. Edukasi dan Penelit. Inform.*, 2021, [Online]. Available: www.kompas.com.

[12] I. F. Rozi, K. S. Batubulan, and M. Rusbandi, "Otomatisasi Peringkasan Teks Pada Dokumen Hukum Menggunakan Metode *Latent Semantic Analysis*," *JIP (Jurnal Inform. Polinema)* , vol. 7, no. 3, pp. 9–15, 2021.

[13] N. Pamungkas *et al.*, "Comparison of Stemming Test Results of Tala Algorithms with Nazief Adriani in Abstract Documents and National News," *Inf. J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 8, no. 1, 2023, doi: 10.25139/inform.v8i1.5569.

[14] A. Zezen, Z. Abidin, and E. Nurjanah, "Sistem Peringkas Teks Otomatis Multi Dokumen Kliping Artikel Berita Gempa Menggunakan Metode Tf-Idf," 2020.

[15] A. S. Alammary, "Arabic Questions Classification Using Modified TF-IDF," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3094115.

[16] W. Darmalaksana, C. Slamet, W. B. Zulfikar, I. F. Fadillah, D. S. adillah Maylawati, and H. Ali, "*Latent Semantic Analysis* and cosine similarity for hadith search engine," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 1, 2020, doi: 10.12928/TELKOMNIKA.V18I1.14874.