JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 11, No. 1, March 2026, Pp.156-164

# ANALISIS PERBANDINGAN KINERJA ALGORITMA STEMMING NAZIEF-ADRIANI DAN IN-IDRIS DALAM PENGOLAHAN TEKS BAHASA INDONESIA

**Muhammad Iqbal *[1)], Ema Utami [2)], Anggit Dwi Hartanto [3)]**
1. Teknik Informatika, Universitas Amikom Yogyakarta, Indonesia
2. Teknik Informatika, Universitas Amikom Yogyakarta, Indonesia
3. Teknik Informatika, Universitas Amikom Yogyakarta, Indonesia

**Article Info**
**Kata Kunci:** Stemming, Nazief-Adriani, In-Idris, NLP, Bahasa Indonesia

**Keywords:** Stemming, Nazief-Adriani, In-Idris, NLP, Indonesian language

**Article history:**
Received 25 April 2025
Revised 5 June 2025
Accepted 7 July 2025
Available online 1 May 2026

**DOI :**
https://doi.org/10.29100/jipi.v11i1.7723

* Corresponding author.
Corresponding Author
E-mail address:
miqbal20@students.amikom.ac.id

**ABSTRAK**
*Stemming* merupakan salah satu tahap penting dalam pemrosesan bahasa alami (Natural Language Processing) untuk mengubah kata berimbuhan menjadi bentuk dasarnya. Penelitian ini membandingkan dua algoritma *stemming* populer dalam Bahasa Indonesia, yaitu Nazief-Adriani yang berbasis kamus dan In-Idris yang berbasis aturan. Evaluasi dilakukan terhadap tiga jenis dokumen dengan karakteristik gaya bahasa berbeda untuk mengukur kinerja masing-masing algoritma berdasarkan tiga parameter: akurasi, durasi pemrosesan, dan *Root Mean Square Error* (RMSE). Hasil menunjukkan bahwa algoritma In-Idris memiliki tingkat akurasi lebih tinggi (rata-rata 86%) dan RMSE lebih rendah (0.13) dibandingkan Nazief-Adriani (75,3% dan 0.24), sehingga lebih stabil dan akurat dalam mengidentifikasi bentuk dasar kata. Sementara itu, Nazief-Adriani menunjukkan keunggulan dalam efisiensi waktu pemrosesan. Temuan ini menegaskan pentingnya pemilihan algoritma yang disesuaikan dengan kebutuhan spesifik aplikasi NLP, serta membuka peluang pengembangan pendekatan hibrida untuk hasil yang lebih optimal.

**ABSTRACT**
Stemming is a crucial step in Natural Language Processing (NLP) to reduce inflected words to their base forms. This study compares two prominent stemming algorithms for the Indonesian language: the dic-tionary-based Nazief-Adriani and the rule-based In-Idris. Evaluation was conducted on three types of documents with varying linguistic styles to assess their performance based on three parameters: accuracy, processing time, and Root Mean Square Error (RMSE). The results show that In-Idris consistently outperforms Nazief-Adriani in terms of accuracy (aver-age 86%) and RMSE (0.13), indicating more consistent and precise base word identification. On the other hand, Nazief-Adriani demonstrates better efficiency in processing time. These findings highlight the im-portance of selecting an algorithm that aligns with the specific needs of NLP applications and suggest potential for developing hybrid approach-es to achieve more optimal results.

.

## I. PENDAHULUAN

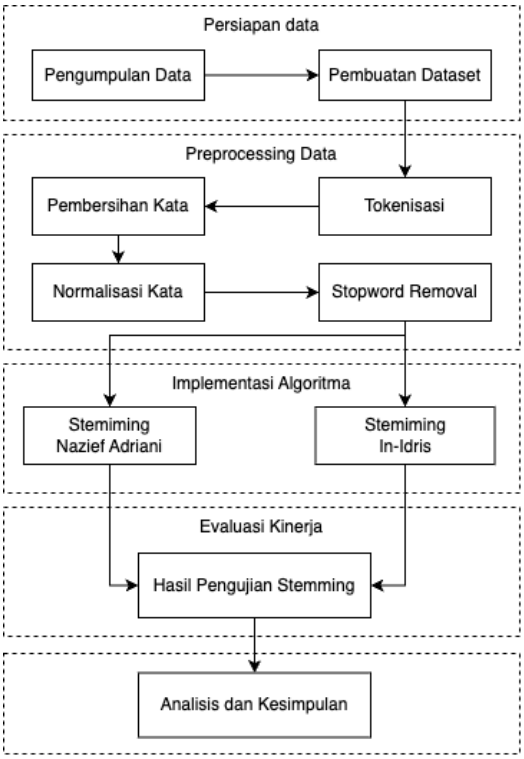
BAHASA Indonesia memiliki karakteristik morfologis yang kompleks yang membedakannya dari banyak bahasa lain, terutama dalam konteks pemrosesan bahasa alami (NLP). Struktur kata dalam Bahasa Indonesia dibentuk melalui kombinasi prefiks, sufiks, infiks, dan konfiks, yang memungkinkan satu kata dasar membentuk puluhan turunan kata dengan makna yang dapat berubah secara signifikan. Tantangan ini tidak banyak ditemui dalam bahasa seperti Inggris yang lebih linear dan analitik dalam pembentukan katanya. Kompleksitas morfologi ini berdampak langsung pada performa model NLP, khususnya dalam tahap *stemming* yang bertugas mengembalikan kata ke bentuk dasarnya sebelum dianalisis lebih lanjut [1][2]. Jika *stemming* tidak akurat, maka hasil proses selanjutnya baik klasifikasi teks, analisis emosi, maupun pencarian informasi akan terdampak secara signifikan. Berdasarkan kompleksitas tersebut, penelitian ini memilih membandingkan algoritma Nazief-Adriani dan In-Idris karena keduanya merupakan pendekatan paling banyak digunakan dalam pemrosesan teks berbahasa Indonesia. Nazief-Adriani dikenal luas dalam komunitas NLP lokal karena pendekatannya yang berbasis kamus dan telah diimplementasikan dalam berbagai aplikasi praktis seperti chatbot dan sistem deteksi

kesalahan penulisan [3][4]. Sementara itu, In-Idris merupakan pengembangan dari algoritma Idris yang awalnya dirancang untuk bahasa Melayu dan dimodifikasi agar lebih sesuai dengan struktur bahasa Indonesia yang serumpun. Pendekatan berbasis aturan yang digunakan In-Idris terbukti lebih adaptif terhadap variasi morfologis yang kompleks [5]. Dibandingkan algoritma *stemming* lain seperti Porter atau Paice-Husk yang dikembangkan untuk bahasa Inggris, kedua algoritma ini lebih kontekstual dan relevan bagi NLP Bahasa Indonesia [6]. Efektivitas *stemming* sangat memengaruhi kualitas layanan berbasis NLP dalam implementasi dunia nyata. Dalam sistem chatbot, kesalahan *stemming* dapat menyebabkan salah pemahaman intent pengguna, sementara dalam sistem pencarian informasi, kata yang tidak diubah ke bentuk dasar dengan benar dapat mengurangi relevansi hasil pencarian [7]. Selain itu, pada aplikasi analisis sentimen, bentuk kata yang tidak tepat dapat memengaruhi interpretasi emosional dalam teks, sehingga memunculkan bias dalam klasifikasi. Di sisi lain, efisiensi waktu pemrosesan juga menjadi faktor kritis, terutama untuk aplikasi real-time seperti layanan pelanggan otomatis, asisten virtual, atau sistem pencarian cepat [8]. Oleh karena itu, memahami bagaimana masing-masing algoritma bekerja dalam konteks akurasi dan efisiensi menjadi landasan penting dalam pengembangan sistem NLP yang handal. Kontribusi utama dari penelitian ini adalah mengisi kesenjangan dalam literatur yang selama ini hanya membandingkan Nazief-Adriani dengan algoritma asing atau membahas In-Idris secara terbatas. Studi ini menyajikan pendekatan evaluasi yang lebih komprehensif dengan menggunakan tiga parameter kuantitatif sekaligus akurasi, waktu pemrosesan, dan *Root Mean Square Error* (RMSE) yang jarang dipertimbangkan secara bersamaan dalam studi sebelumnya [8][9]. Selain itu, penggunaan tiga jenis dokumen dengan gaya linguistik berbeda (jurnalistik, sastra, dan hukum) memberikan validitas eksternal yang lebih kuat karena mencerminkan variasi penggunaan bahasa dalam konteks yang nyata dan heterogen. Temuan penelitian ini tidak hanya memberikan wawasan praktis dalam pemilihan algoritma *stemming* berdasarkan kebutuhan spesifik aplikasi, tetapi juga membuka peluang pengembangan lebih lanjut di bidang NLP Bahasa Indonesia. Salah satu arah penting yang disarankan adalah pengembangan algoritma hybrid yang menggabungkan keunggulan pendekatan berbasis kamus (efisiensi waktu) dengan pendekatan berbasis aturan (akurasi dalam menangani morfologi kompleks). Lebih jauh lagi, integrasi dengan pendekatan pembelajaran mesin seperti deep learning atau model transformer berpotensi meningkatkan kemampuan generalisasi terhadap bahasa informal dan kontekstual seperti yang sering ditemukan di media sosial [1][3][9]. Dengan demikian, hasil penelitian ini memberikan fondasi penting dalam mendorong kemajuan teknologi bahasa Indonesia di masa mendatang.

## II. METODE PENELITIAN

Penelitian ini menggunakan metode kuantitatif eksperimental untuk menguji dan membandingkan kinerja algoritma *stemming* Nazief-Adriani dan In-Idris dalam pemrosesan teks bahasa Indonesia. Pendekatan eksperimental ini dilakukan dengan cara sistematis melalui penerapan kedua algoritma pada dataset yang telah dikumpulkan, diikuti dengan evaluasi performa berdasarkan serangkaian metrik kuantitatif yang telah ditentukan sebelumnya. Metode evaluasi berbasis eksperimen ini telah terbukti efektif dalam penelitian NLP untuk mengukur kehandalan algoritma dalam tugas-tugas spesifik [1]. Keunggulan pendekatan ini terletak pada kemampuannya memberikan kendali ketat terhadap variabel input sekaligus memastikan evaluasi yang objektif dan terukur. Metodologi serupa telah diimplementasikan dalam studi-studi berbasis media sosial dan korpus domain campuran untuk menilai ketahanan algoritma ketika menghadapi variasi kompleks bahasa informal maupun formal [10][11]. Untuk memastikan validitas eksternal yang tinggi, penelitian ini secara khusus menekankan pentingnya pengujian komprehensif terhadap berbagai jenis teks, sehingga hasil temuan dapat digeneralisasi ke aplikasi dunia nyata seperti klasifikasi berita, pencarian dokumen hukum, dan analisis sentimen di platform media sosial [12]. Dengan mempertimbangkan semua aspek tersebut, metode kuantitatif eksperimental dipilih sebagai pendekatan yang paling tepat untuk memberikan gambaran menyeluruh tentang efektivitas masing-masing algoritma dalam konteks pemrosesan bahasa Indonesia. Visualisasi tahapan metodologis dalam penelitian ini dapat dilihat pada gambar di bawah ini, yang menggambarkan alur kerja sistematis dari pengumpulan data hingga analisis hasil.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 11, No. 1, March 2026, Pp.156-164




Gambar 1 Alur Penelitian

### *A. Persiapan Data*

Data teks yang digunakan dalam penelitian ini terdiri atas kumpulan teks berbahasa Indonesia yang diperoleh dari berbagai sumber, seperti korpus berita, dokumen resmi, dan artikel. Keberagaman data ini bertujuan untuk memastikan bahwa algoritma diuji pada berbagai variasi kata dan struktur kalimat. Selanjutnya, data yang telah dikumpulkan diseleksi dan diubah menjadi dataset, sesuai dengan rekomendasi dalam penelitian terbaru mengenai metode stemming dalam NLP [1].

### *B. Preprocessing Data*

Data teks diambil dari berbagai sumber seperti korpus berita, dokumen akademik, dan media sosial untuk menjamin keragaman struktur kalimat dan kosa kata. Pendekatan ini selaras dengan metode dalam studi yang menguji kinerja algoritma pada dokumen dengan gaya linguistik yang berbeda [2], serta pendekatan serupa pada pengujian algoritma dalam klasifikasi dokumen [13]. Sebelum dilakukan *stemming*, data akan melalui tahap preprocessing agar siap digunakan, prosesnya meliputi:

- **Tokenisasi**: Memisahkan teks menjadi kata-kata individu.
- **Pembersihan Kata**: Menghilangkan tanda baca, angka, dan simbol yang tidak diperlukan.
- **Normalisasi Kata**: Mengubah kata-kata ke dalam format yang seragam (misalnya, mengubah ejaan tidak baku men-jadi baku).
- **Stopword Removal**: Menghapus kata-kata umum yang tidak memiliki makna signifikan dalam analisis teks (seperti “yang”, “dan”, “di”, dll.).

Metode ini mengikuti pendekatan yang telah digunakan dalam penelitian sebelumnya yang menunjukkan bahwa *preprocessing* yang tepat dapat meningkatkan akurasi model NLP [2][6].

### *C. Implementasi Algoritma Stemming*

Pada tahap ini, algoritma Nazief-Adriani dan In-Idris akan diterapkan pada dataset yang telah diproses sebelumnya. Setiap kata dalam dataset akan dikembalikan ke bentuk dasarnya menggunakan masing-masing algoritma:

- **Nazief-Adriani**: menggunakan pendekatan kamus dan rekursif yang banyak digunakan dalam aplikasi nyata, termasuk chatbot dan sistem pendeteksi kesalahan [4].
- **In-Idris**: algoritma berbasis aturan yang dimodifikasi dari sistem bahasa Melayu dan terbukti efektif pada teks Bahasa Indonesia dengan struktur kompleks [5].

Pendekatan ini sesuai dengan penelitian terdahulu yang membandingkan berbagai metode *stemming* dan menunjukkan bahwa algoritma berbasis aturan sering kali lebih efisien dalam bahasa dengan morfologi kompleks.

### D. *Evaluasi Kinerja Algoritma*

Tahap pengujian dilakukan untuk mengevaluasi performa algoritma dalam hal akurasi dan efisiensi. Pengujian dilakukan dengan memproses dataset menggunakan algoritma Nazief-Adriani dan In-Idris. Parameter pengujiannya antara lain:

*1) Akurasi*

Mengukur persentase kata yang berhasil di-stemming ke bentuk dasar yang benar [6].

$$Accuracy = \frac{Correct\ Stemming\ Results}{Total\ Words} \times 100\%$$

*2) Efisiensi Waktu*

Mengukur waktu yang diperlukan algoritma untuk memproses seluruh dataset [14].

*3) Root Mean Square Error (RMSE)*

Mengukur ukuran galat prediksi dari bentuk dasar yang ideal [3].

Evaluasi ini didasarkan pada metrik yang digunakan dalam penelitian sebelumnya yang meneliti efektivitas algoritma stemming dalam berbagai bahasa [15].

## III. HASIL DAN PEMBAHASAN

Penelitian ini membandingkan kinerja dua algoritma *stemming* bahasa Indonesia, yaitu Nazief-Adriani dan In-Idris, dengan menerapkan keduanya pada tiga dokumen yang memiliki karakteristik linguistik yang berbeda. Evaluasi dilakukan berdasarkan tiga parameter utama, yaitu akurasi stemming, durasi pemrosesan (kecepatan), dan *Root Mean Square Error* (RMSE). Hasil pengujian ditampilkan dalam bentuk tabel dan grafik untuk mendukung analisis kuantitatif:

### A. *Objek Pengujian*

Penelitian ini pada tahap awal menggunakan dokumen berbahasa Indonesia sebagai objek uji algoritma *stemming*. Doku-men dipilih secara strategis untuk mencerminkan variasi struktur teks, gaya bahasa, dan tingkat kompleksitas morfologis. Pemilihan ini bertujuan untuk menguji generalisasi algoritma pada berbagai konteks teks Bahasa Indonesia [1]. Ketiga dokumen yang digunakan sebagai objek pengujian merupakan teks berbahasa Indonesia dengan gaya linguistik yang berbeda: artikel populer, esai sastra, dan dokumen hukum formal. Dokumen tersebut dapat diakses secara publik melalui tautan berikut:

1) Preservasi Bahasa (Language Preservation) – artikel dari Badan Pengembangan dan Pembinaan Bahasa: https://badanbahasa.kemdikbud.go.id/artikel-detail/4476/preservasi-bahasa-language-preservation
2) Memanusiakan Indonesia dengan Bahasa Sastrawi – esai dari Badan Pengembangan dan Pembinaan Bahasa: https://badanbahasa.kemdikbud.go.id/artikel-detail/4473/memanusiakan-indonesia-dengan-bahasa-sastrawi
3) Pembukaan UUD 1945 – dokumen hukum dari Mahkamah Konstitusi Republik Indonesia: https://www.mkri.id/public/content/infoumum/regulation/pdf/UUD45%20ASLI.pdf

### B. *Hasil Kuantitatif*

Evaluasi kuantitatif dilakukan dengan menggunakan aplikasi *stemming* berbasis web yang dikembangkan secara mandiri oleh peneliti. Sistem ini terdiri dari antarmuka pengguna berbasis Nuxt.js dan *backend* berbasis FastAPI, yang memungkinkan pengujian algoritma dilakukan secara langsung melalui REST API. Dalam sistem ini, pengguna dapat mengunggah dokumen atau mengetik teks secara langsung, kemudian memilih algoritma *stemming* yang akan digunakan, baik Nazief-Adriani maupun In-Idris. Backend akan mengeksekusi proses stemming dan mengembalikan hasil beserta data evaluasi, seperti akurasi prediksi kata dasar, durasi pemrosesan, dan *Root Mean Square Error* (RMSE). Dengan demikian, pengujian dilakukan dalam kondisi yang merepresentasikan lingkungan aplikasi nyata, termasuk mempertimb;'angkan latensi jaringan, waktu proses API, serta beban parsing dan *preprocessing* secara aktual. Pendekatan ini memberikan keunggulan dibandingkan simulasi lokal karena menggambarkan performa algoritma dalam konteks sistem NLP produksi seperti chatbot atau analisis dokumen hukum. Hasil pengujian menunjukkan bahwa algoritma In-Idris mencatat akurasi rata-rata sebesar 86,0%, sedangkan Nazief-Adriani sebesar 75,3%. Dari sisi waktu pemrosesan, Nazief-Adriani unggul dengan durasi rata-rata 2,49 milidetik per kata, sementara In-Idris membutuhkan 3,46 milidetik. Hal ini sejalan dengan karakteristik arsitektural masing-masing: Nazief-Adriani yang berbasis kamus lebih ringan, sedangkan In-Idris yang berbasis

aturan memerlukan evaluasi morfologis yang lebih kompleks. Dari aspek kestabilan keluaran, algoritma In-Idris juga lebih unggul dengan nilai RMSE sebesar 0,13 dibandingkan 0,24 pada Nazief-Adriani. Nilai RMSE yang lebih rendah menandakan bahwa In-Idris menghasilkan bentuk dasar yang lebih konsisten terhadap referensi manual, khususnya pada kata-kata dengan struktur imbuhan kompleks atau bentuk tidak baku. Tabel berikut menyajikan perbandingan nilai kuantitatif dari Akurasi dan Durasi. Evaluasi numerik seperti ini penting untuk membandingkan performa algoritma secara objektif dan telah diterapkan luas dalam kajian NLP dan *information retrieval* [16][8].

TABEL I
HASIL PERBANDINGAN AKURASI

| No | Dokumen | Jumlah Kata | Akurasi In-Idris | Akurasi Nazief & Adriani |
|---|---|---|---|---|
| 1 | Artikel Preservasi Bahasa | 1888 | 87.18% | 75.79% |
| 2 | Artikel Memanusiakan Indonesia dengan Bahasa Sastrawi | 719 | 88.18% | 76.77% |
| 3 | Pembukaan UUD 1945 | 2320 | 84.61% | 73.36% |

TABEL 2
HASIL PERBANDINGAN DURASI

| No | Dokumen | Jumlah Kata | Durasi (m/s) In-Idris | Durasi (m/s) Nazief & Adriani |
|---|---|---|---|---|
| 1 | Artikel Preservasi Bahasa | 1888 | 2.357 | 1.264 |
| 2 | Artikel Memanusiakan Indonesia dengan Bahasa Sastrawi | 719 | 0.832 | 0.706 |
| 3 | Pembukaan UUD 1945 | 2320 | 7.190 | 5.514 |

### C. *Analisis Akurasi*

Hasil evaluasi menunjukkan bahwa algoritma In-Idris memiliki akurasi rata-rata sebesar 86,0%, lebih tinggi dibandingkan Nazief-Adriani yang mencatat 75,3%. Perbedaan sebesar hampir 11% ini memiliki implikasi praktis yang signifikan dalam berbagai aplikasi NLP. Dalam sistem klasifikasi teks, analisis sentimen, dan ekstraksi informasi, perbedaan akurasi sebesar ini dapat menyebabkan perbedaan interpretasi terhadap konten yang berdampak langsung pada kualitas hasil. Misalnya, pada tugas klasifikasi opini publik atau dokumen hukum, *stemming* yang tidak akurat dapat mengakibatkan kesalahan dalam identifikasi tema atau entitas utama dalam teks. Penelitian oleh Pradana dan Hayaty (2020) menunjukkan bahwa performa *stemming* yang tidak optimal dapat menurunkan akurasi klasifikasi sentimen hingga 15% dalam korpus bahasa Indonesia [17]. Selain itu, penelitian oleh Rizki (2023) pada data media sosial Twitter membuktikan bahwa meskipun algoritma Nazief-Adriani mampu mencapai akurasi tinggi (hingga 90,34%) pada data formal, performanya menurun signifikan pada teks yang mengandung campuran bahasa atau bentuk morfologi informal [2]. Temuan ini menggarisbawahi pentingnya memilih algoritma yang mampu menangani variasi bahasa dan struktur kata tidak baku secara fleksibel, sebagaimana dilakukan oleh In-Idris yang berbasis aturan. Lebih jauh, penelitian oleh Ramdhani et al. (2020) dalam klasifikasi berita menggunakan CNN menunjukkan bahwa peningkatan akurasi *stemming* secara signifikan meningkatkan kinerja model klasifikasi, khususnya pada label yang bergantung pada kata kerja atau kata sifat berimbuhan [18]. Sementara itu, Rianto et al. (2021) menekankan bahwa *stemming* yang dirancang untuk konteks informal dapat meningkatkan akurasi klasifikasi teks non-formal dari 73% menjadi 85%, menunjukkan dampak besar dari pemilihan metode *stemming* yang tepat terhadap efektivitas keseluruhan sistem NLP [11]. Secara keseluruhan, temuan ini memperkuat bahwa algoritma In-Idris lebih unggul dalam menghadapi kompleksitas struktur morfologi bahasa Indonesia, khususnya pada teks dengan tingkat variasi linguistik tinggi. Hal ini menjadikannya kandidat yang lebih andal untuk aplikasi yang membutuhkan akurasi semantik tinggi, sementara Nazief-Adriani tetap relevan untuk tugas-tugas dengan kebutuhan kecepatan tinggi dan struktur bahasa yang lebih baku.

### D. *Analisis Efisiensi Waktu*

Efisiensi waktu pemrosesan merupakan komponen krusial dalam penilaian kinerja algoritma NLP, terutama ketika algoritma tersebut diterapkan dalam sistem yang menuntut kecepatan tinggi seperti chatbot, sistem pencarian cerdas, atau aplikasi berbasis IoT. Hasil penelitian menunjukkan bahwa Nazief-Adriani lebih efisien, dengan rata-rata waktu pemrosesan 2,49 milidetik per kata, dibandingkan dengan In-Idris yang mencatat 3,46 milidetik per kata. Hal ini disebabkan oleh arsitektur Nazief-Adriani yang mengandalkan pencocokan kamus langsung, sedangkan In-Idris harus menjalankan analisis aturan morfologis bertingkat yang lebih kompleks. Dalam konteks real-time, keunggulan ini menjadikan Nazief-Adriani lebih sesuai untuk aplikasi yang menuntut latensi rendah dan volume trafik tinggi. Temuan ini didukung oleh studi Sasmal (2023) yang menunjukkan bahwa dalam sistem pemrosesan data real-time, pemilihan algoritma yang lebih ringan secara komputasional dapat secara signifikan

meningkatkan throughput dan efisiensi sistem, meskipun kadang dengan sedikit pengorbanan akurasi [19]. Dalam penelitian tersebut menekankan bahwa meskipun akurasi penting, pengguna cenderung lebih menghargai kecepatan sistem selama hasilnya masih relevan secara semantik. Namun, kecepatan pemrosesan tidak selalu menjadi prioritas utama. Dalam skenario pemrosesan dokumen hukum, arsip akademik, atau analisis longitudinal kebijakan, akurasi dan konsistensi hasil jauh lebih penting daripada latensi. Studi oleh Riyadi et al. (2024) menunjukkan bahwa pemodelan teks hukum dengan IndoGovBERT lebih memprioritaskan akurasi meskipun berdampak pada waktu pemrosesan, karena kesalahan interpretasi pada dokumen semacam itu dapat berdampak serius pada pengambilan keputusan [20]. Ini menegaskan bahwa In-Idris tetap menjadi pilihan unggul untuk aplikasi yang mengutamakan presisi linguistik, meskipun lebih lambat secara komputasional. Dengan mempertimbangkan hasil ini, dapat disimpulkan bahwa pemilihan algoritma tidak dapat dipisahkan dari konteks aplikasinya. Nazief-Adriani unggul untuk sistem responsif dengan keterbatasan sumber daya dan kebutuhan pemrosesan cepat, sementara In-Idris lebih tepat digunakan dalam sistem analitis atau sektor yang membutuhkan akurasi morfologis tinggi. Pendekatan berbasis konteks ini penting dalam desain arsitektur sistem NLP berbahasa Indonesia agar optimal dalam performa maupun hasil.

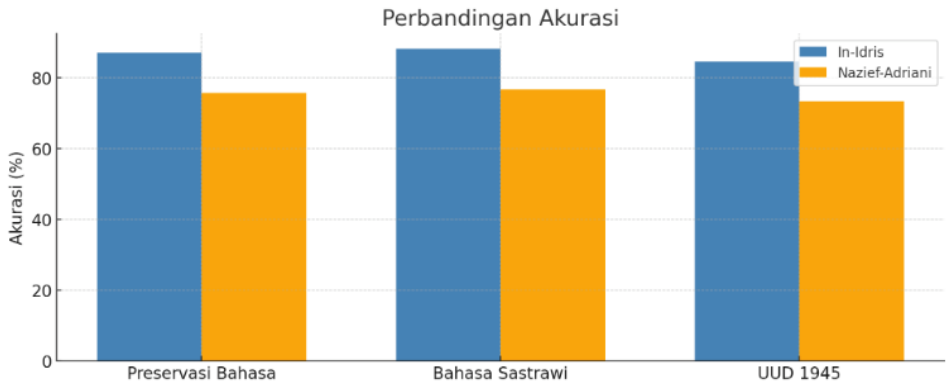
### *E. Evaluasi RMSE dan Analisis pola kesalahan*

Selain akurasi dan efisiensi waktu, metrik *Root Mean Square Error* (RMSE) digunakan dalam penelitian ini untuk menilai konsistensi hasil *stemming* terhadap bentuk dasar referensi. Hasil evaluasi menunjukkan bahwa algoritma In-Idris mencatat RMSE lebih rendah (0,13) dibandingkan dengan Nazief-Adriani (0,24). Nilai RMSE yang lebih rendah menunjukkan bahwa In-Idris menghasilkan prediksi bentuk dasar yang lebih dekat dan stabil terhadap label rujukan, mengurangi kesalahan besar akibat *overstemming* atau *understemming*. Ini sangat penting dalam aplikasi NLP seperti sistem pengarsipan digital atau klasifikasi dokumen hukum, di mana kesalahan morfologis dapat merusak interpretasi semantik dari suatu teks. Pola kesalahan utama yang ditemukan pada Nazief-Adriani mencakup *overstemming* pada kata-kata berimbuhan kompleks seperti "memperjuangkan" yang sering dipotong menjadi "juang" alih-alih bentuk dasar yang tepat "perjuang" atau "ajar" menjadi "ja". Pendekatan berbasis kamus tidak mampu menangani kombinasi afiks yang tidak tersedia dalam daftar, terutama jika bentuk kata tersebut jarang digunakan atau domain-spesifik. Sebaliknya, In-Idris yang berbasis aturan morfologis menunjukkan performa lebih baik dalam menghadapi kata-kata dengan imbuhan bertingkat karena proses pemangkasan mengikuti struktur sistematis berdasarkan pola pembentukan kata. Studi oleh Putra et al. (2020) dalam konteks penerapan *stemming* pada teks keagamaan berbahasa Indonesia (terjemahan Quran) juga mencatat bahwa penggunaan aturan bertingkat dalam pemrosesan kata dasar mampu mengurangi kesalahan *stemming* hingga 40% dibandingkan pendekatan konvensional [21]. Selain itu, penelitian oleh Kharis et al. (2025) yang membandingkan performa beberapa model prediksi berbasis teks dengan evaluasi RMSE menunjukkan bahwa galat rendah berkorelasi kuat dengan kualitas *preprocessing*, termasuk *stemming*, yang stabil dan akurat [22]. Secara keseluruhan, metrik RMSE memberikan dimensi tambahan dalam mengevaluasi algoritma *stemming* karena dapat menangkap variasi kesalahan, bukan hanya hitungan benar atau salah seperti dalam akurasi biasa. Oleh karena itu, nilai RMSE yang lebih rendah pada In-Idris memperkuat argumen bahwa algoritma ini lebih cocok untuk aplikasi yang memerlukan tingkat konsistensi linguistik tinggi, dan membuka ruang untuk integrasi sistem koreksi galat berbasis pembelajaran mesin untuk memperbaiki pola kesalahan yang berulang.

### *F. Visualisasi Hasil*

Berikut adalah grafik batang yang menunjukkan perbandingan akurasi, durasi, dan RMSE antara kedua algoritma:

1. Grafik Akurasi

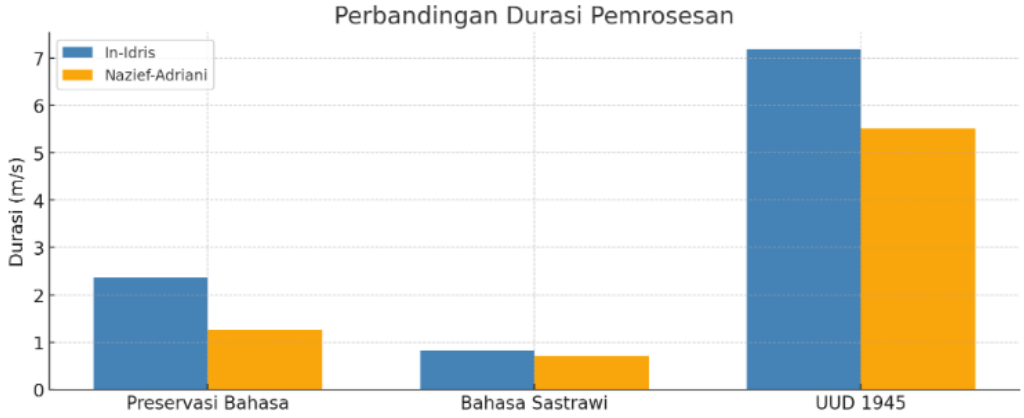



Gambar 2 Grafik Perbandingan Akurasi

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 11, No. 1, March 2026, Pp.156-164

Gambar di atas menyajikan perbandingan kinerja dua algoritma *stemming* Bahasa Indonesia—Nazief-Adriani dan In-Idris—berdasarkan tiga metrik utama: akurasi, durasi pemrosesan, dan *Root Mean Square Error* (RMSE). Pada grafik pertama, terlihat bahwa In-Idris memiliki akurasi lebih tinggi, yaitu sebesar 86,0%, dibandingkan Nazief-Adriani yang mencatat 75,3%. Perbedaan ini mendukung temuan sebelumnya yang menunjukkan bahwa pendekatan berbasis aturan lebih unggul dalam menangani struktur morfologis kompleks Bahasa Indonesia, terutama dalam konteks teks non-baku atau kreatif seperti media sosial dan esai sastra [17].

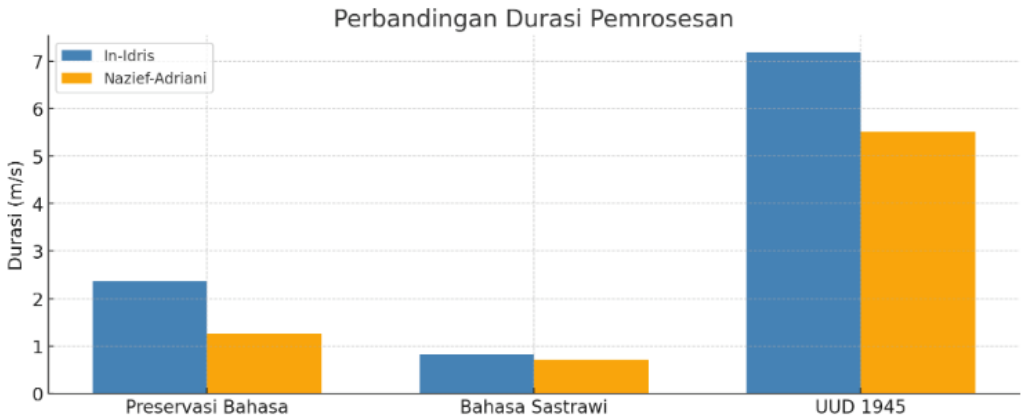
2. Grafik Durasi




Gambar 3 Grafik Perbandingan Durasi

Grafik diatas menunjukkan durasi pemrosesan rata-rata per kata, di mana Nazief-Adriani lebih unggul dalam efisiensi waktu dengan waktu rata-rata 2,49 milidetik, sedangkan In-Idris membutuhkan 3,46 milidetik. Hal ini sesuai dengan karakteristik arsitektural algoritma—Nazief-Adriani yang berbasis kamus hanya melakukan pencocokan sederhana, sedangkan In-Idris membutuhkan evaluasi aturan morfologis yang lebih dalam. Efisiensi dalam sistem real-time sangat bergantung pada kompleksitas algoritma, dan algoritma dengan struktur linier biasanya lebih cepat meski sedikit mengorbankan akurasi [19].

3. Grafik RMSE




Gambar 4 Grafik Perbandingan RMSE

Grafik ketiga memperlihatkan nilai RMSE, yang mencerminkan stabilitas hasil *stemming* terhadap bentuk dasar referensi. In-Idris mencatat nilai RMSE lebih rendah (0,13) dibandingkan Nazief-Adriani (0,24), yang menandakan prediksi kata dasar lebih konsisten. Pendekatan berbasis aturan memungkinkan In-Idris mengenali struktur imbuhan yang kompleks secara sistematis. Paturan morfologis dalam sistem *stemming* dapat secara signifikan mengurangi kesalahan pemotongan akar kata, terutama pada teks domain tertentu seperti keagamaan atau hukum [21].

Secara keseluruhan, visualisasi ini menunjukkan adanya *trade-off* antara kecepatan dan ketepatan. Nazief-Adriani cocok untuk aplikasi yang menuntut respon cepat dan volume tinggi, seperti chatbot atau pencarian teks

cepat. Di sisi lain, In-Idris unggul dalam akurasi dan konsistensi, menjadikannya ideal untuk aplikasi yang memerlukan ketelitian linguistik, seperti klasifikasi dokumen hukum atau sistem rekomendasi berbasis konteks. kualitas *preprocessing* yang konsisten, termasuk *stemming*, memiliki pengaruh signifikan terhadap performa akhir model NLP berbasis bahasa Indonesia [22].

## IV. KESIMPULAN

Penelitian ini membandingkan dua algoritma *stemming* Bahasa Indonesia, yakni Nazief-Adriani dan In-Idris, dengan menguji performanya pada tiga jenis dokumen yang mewakili variasi gaya linguistik: jurnalistik, sastra, dan hukum. Evaluasi dilakukan berdasarkan tiga parameter utama, yaitu akurasi, durasi pemrosesan, dan *Root Mean Square Error* (RMSE). Hasil menunjukkan bahwa:

1) In-Idris memiliki akurasi lebih tinggi secara konsisten dibandingkan Nazief-Adriani, dengan rata-rata 86,0% berbanding 75,3%, menunjukkan kemampuannya dalam menangani struktur morfologis kompleks secara lebih adaptif.
2) Dari segi efisiensi waktu, Nazief-Adriani lebih unggul, terutama dalam pemrosesan dokumen berdurasi pendek hingga menengah, dengan waktu rata-rata 2,49 ms per kata dibandingkan 3,46 ms pada In-Idris.
3) Dalam aspek RMSE, In-Idris menunjukkan nilai galat yang lebih rendah (0,13) dibandingkan Nazief-Adriani (0,24), yang mengindikasikan hasil *stemming* yang lebih konsisten terhadap bentuk dasar yang benar.

Temuan ini menunjukkan bahwa In-Idris lebih sesuai untuk aplikasi NLP yang mengutamakan akurasi dan kestabilan hasil, seperti klasifikasi dokumen formal, ekstraksi informasi hukum, dan sistem pengolahan korpus akademik. Di sisi lain, Nazief-Adriani tetap relevan untuk aplikasi real-time seperti chatbot, sistem pencarian cepat, dan rekomendasi teks, di mana efisiensi waktu menjadi prioritas utama. Sebagai arah pengembangan lebih lanjut, disarankan untuk membangun algoritma hybrid yang menggabungkan efisiensi pencocokan kamus Nazief-Adriani dengan fleksibilitas aturan morfologis In-Idris. Pendekatan ini dinilai menjanjikan dalam mengoptimalkan performa dari sisi akurasi dan efisiensi secara bersamaan. Selain itu, penelitian di masa mendatang perlu menguji algoritma ini pada dataset yang lebih besar dan heterogen, termasuk teks-teks informal seperti komentar media sosial, forum diskusi, maupun bahasa lokal, guna mengukur kemampuan generalisasi sistem NLP dalam konteks linguistik yang dinamis. Sejalan dengan tren terbaru, integrasi algoritma *stemming* dengan model pembelajaran mesin seperti transformer-based language models (misalnya IndoBERT) juga dapat menjadi solusi jangka panjang untuk meningkatkan adaptabilitas sistem terhadap keragaman bahasa Indonesia kontemporer [22][20].

## DAFTAR PUSTAKA

[1] A. Jabbar, S. Iqbal, M. I. Tamimy, A. Rehman, S. A. Bahaj, and T. Saba, "An Analytical Analysis of Text Stemming Methodologies in Information Retrieval and Natural Language Processing Systems," *IEEE Access*, vol. 11, no. December, pp. 133681–133702, 2023, doi: 10.1109/ACCESS.2023.3332710.

[2] A. S. Rizki, N. M. Aristi, N. Ridha, A. F. Zulfahri, and D. A. Wibowo, "Implementation of The Indonesian Language Stemming Algorithm in Twitter Data Preprocessing. Case Study: Twitter Wargabanua and Instakalsel," *Fidel. J. Tek. Elektro*, vol. 5, no. 3, pp. 175–183, 2023, doi: 10.52005/fidelity.v5i3.170.

[3] M. Fikri and Z. Abidin, "Analysis of the Use of Nazief-Adriani Stemming and Porter Stemming in COVID-19 Twitter Sentiment Analysis with Term Frequency- Inverse Document Frequency Weighting Based on k-Nearest Neighbor Algorithm," vol. 2, no. 2, pp. 80–87, 2024, doi: 10.15294/rji.v2i2.74267.

[4] A. Yudhana, A. Fadlil, and M. Rosidin, "Indonesian words error detection system using nazief adriani stemmer algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 12, pp. 219–225, 2019, doi: 10.14569/ijacsa.2019.0101231.

[5] F. W. Suci, N. Hayatin, and Y. Munarko, "in-Idris: Modification of Idris Stemming Algorithm for Indonesian Text," *IIUM Eng. J.*, vol. 23, no. 1, pp. 82–94, 2022, doi: 10.31436/IIUMEJ.V23I1.1783.

[6] D. Mustikasari, I. Widaningrum, R. Arifin, and W. H. E. Putri, "Comparison of Effectiveness of Stemming Algorithms in Indonesian Documents," *Proc. 2nd Borobudur Int. Symp. Sci. Technol. (BIS-STE 2020)*, vol. 203, pp. 154–158, 2021, doi: 10.2991/aer.k.210810.025.

[7] I. Prismana, D. Prehanto, D. Dermawan, A. Herlingga, and S. Wibawa, "Nazief & Adriani Stemming Algorithm With Cosine Similarity Method For Integrated Telegram Chatbots With Service," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1125, no. 1, p. 012039, 2021, doi: 10.1088/1757-899x/1125/1/012039.

[8] N. Yusliani, R. Primartha, and M. Diana, "Multiprocessing Stemming: A Case Study of Indonesian Stemming," *Int. J. Comput. Appl.*, vol. 182, no. 40, pp. 15–19, 2019, doi: 10.5120/ijca2019918476.

[9] C. Moral, A. de Antonio, R. Imbert, and J. Ramírez, "A survey of stemming algorithms in information retrieval," *Inf. Res.*, vol. 19, no. 1, 2014, doi: 10.1145/3715120.

[10] Erlina, R. G. Pratama Simamora, H. Rasjid, and B. L. Basyah, "Aplikasi Chatbot Online Untuk Pemesanan Tiket Bioskop Menggunakan Natural Language Processing," *J. Nas. Teknol. Komput.*, vol. 3, no. 3, pp. 127–141, 2023, doi: 10.61306/jnastek.v3i3.84.

[11] Rianto, A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, "Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation," *J. Big Data*, vol. 8, pp. 1–16, 2021.

[12] S. J. P. Et.al, "Optimizing Text Categorization for Indonesian Text Using Clustering Label Technique," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 3, pp. 1483–1491, 2021, doi: 10.17762/turcomat.v12i3.947.

[13] B. Ansor *et al.*, "NAKNN : An Efficient Classification of Indonesian News Texts with Nazief-Adriani and KNN," vol. 5, no. 2, pp. 57–64, 2024, doi: 10.26714/jichi.v5i2.15420.

[14] S. Tuhpatussania, E. Utami, and A. D. Hartanto, "Comparison of Porters Stemming Algorithm and Nazief & Adriani'S Stemming Algorithm in

Determining Indonesian Language Learning Modules," *J. Pilar Nusa Mandiri*, vol. 18, no. 2, pp. 203–210, 2022, doi: 10.33480/pilar.v18i2.3940.
[15] 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, pp. 276–289, 2023, doi: 10.29407/intensif.v7i2.20103.
[16] A. Prasidhatama and K. M. Suryaningrum, "Perbandingan Algoritma Nazief & Adriani Dengan Algoritma Idris Untuk Pencarian Kata Dasar," *J. Teknol. dan Manaj. Inform.*, vol. 4, no. 1, pp. 1–4, 2018, doi: 10.26905/jtmi.v4i1.1773.
[17] A. W. Pradana and M. Hayaty, "The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 3, pp. 375–380, 2019, doi: 10.22219/kinetik.v4i4.912.
[18] M. A. Ramdhani, M. A. Ramdhani, D. S. adillah Maylawati, and T. Mantoro, "Indonesian news classification using convolutional neural network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 2, pp. 1000–1009, 2020, doi: 10.11591/ijeecs.v19.i2.pp1000-1009.
[19] S. Sasmal, "Real-time Data Processing with Machine Learning Algorithms," *Int. Res. J. Eng. Appl. Sci.*, vol. 11, no. 4, pp. 91–96, 2023, doi: 10.55083/irjeas.2023.v11i04012.
[20] A. Riyadi, M. Kovacs, U. Serdült, and V. Kryssanov, "IndoGovBERT: A Domain-Specific Language Model for Processing Indonesian Government SDG Documents," *Big Data Cogn. Comput.*, vol. 8, no. 11, pp. 1–28, 2024, doi: 10.3390/bdcc8110153.
[21] S. Putra, N. Cahyanti, S. Ratnawati, M. Gunawan, and D. Sari, "The Implementation of Indonesian Stemming System for Indonesian Translation of the Quran," 2020, doi: 10.4108/eai.2-10-2018.2295507.
[22] S. Anastassia *et al.*, "Comparing machine learning models for Indonesia stock market prediction," vol. 38, no. 1, pp. 508–516, 2025, doi: 10.11591/ijeecs.v38.i1.pp508-516.