TESIS

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

disusun oleh

MUHAMMAD IQBAL

22.55.1218

Business Intelligence

FAKULTAS ILMU KOMPUTER

UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2026

TESIS

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

A COMPARATIVE ANALYSIS OF THE NAZIEF-ADRIANI AND IN-INDRIS STEMMING ALGORITHMS FOR INDONESIAN TEXT PROCESSING

Diajukan untuk memenuhi salah satu syarat mencapai derajat Pascasarjana
Program Studi Bussiness Intelligence

disusun oleh

MUHAMMAD IQBAL
22.55.1218
Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2026

HALAMAN PERSETUJUAN

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

A COMPARATIVE ANALYSIS OF THE NAZIEF-ADRIANI AND IN-INDRIS STEMMING ALGORITHMS FOR INDONESIAN TEXT PROCESSING

yang disusun dan diajukan oleh

Muhammad Iqbal

22.55.1218

telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal 07 Agustus 2025

Dosen Pembimbing,

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

HALAMAN PENGESAHAN

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

A COMPARATIVE ANALYSIS OF THE NAZIEF-ADRIANI AND IN-INDRIS STEMMING ALGORITHMS FOR INDONESIAN TEXT PROCESSING

yang disusun dan diajukan oleh

Muhammad Iqbal

22.55.1218

Telah dipertahankan di depan Dewan Penguji
pada tanggal 07 Agustus 2025

Susunan Dewan Penguji

Nama Penguji	Tanda Tangan
Dr. Andi Sunyoto, M.Kom. **NIK. 190302052**	
Hanif Al Fatta, S.Kom., M.Kom., Ph.D. **NIK. 190302096**	
Prof. Dr. Ema Utami, S.Si., M.Kom. **NIK. 190302037**	

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
Tanggal 07 Agustus 2025

DEKAN FAKULTAS ILMU KOMPUTER

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Muhammad Iqbal
NIM : 22.55.1218

Menyatakan bahwa Tesis dengan judul berikut:

Analisis Perbandingan Kinerja Algoritma Stemming Nazief-Adriani Dan In-Idris Dalam Pengolahan Teks Bahasa Indonesia

Dosen Pembimbing: Prof. Dr. Ema Utami, S.Si., 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 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, 07 Agustus 2025

Yang Menyatakan,

Muhammad Iqbal

HALAMAN PERSEMBAHAN

Dengan segala kerendahan hati dan rasa syukur yang mendalam, karya Tesis ini saya persembahkan kepada:

1. Allah SWT, Dzat Maha Pengasih lagi Maha Penyayang, yang telah memberikan nikmat iman, kesehatan, dan kesempatan, serta kekuatan lahir dan batin dalam menyelesaikan perjalanan panjang ini. Segala puji hanya bagi-Mu, ya Allah.
2. Kedua orang tuaku tercinta yang tak pernah lelah mendoakan, mendukung, dan menyayangi tanpa syarat. Terima kasih atas cinta, kesabaran, serta segala pengorbanan yang tak ternilai. Doa dan restu kalian adalah cahaya dalam setiap langkah perjuangan ini.
3. Kakak dan Adik tersayang yang selalu menjadi penyemangat dan pengingat untuk terus berusaha menjadi lebih baik hingga terselesaikannya Tesis ini.
4. Segenap civitas akademika Universitas Amikom Yogyakarta, para dosen, Staf pengajar, dan karyawan yang telah memberikan ilmu, bimbingan, serta dukungan selama menempuh pendidikan hingga terselesaikannya Tesis ini.
5. Semua pihak yang telah membantu dan mendukung dalam penyusunan tesis ini.

Semoga karya Tesis ini tidak hanya menjadi syarat kelulusan, tetapi juga dapat memberikan manfaat dan kontribusi bagi perkembangan ilmu pengetahuan.

Aamiin Ya Rabbal Alamin.

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas limpahan rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan penyusunan tesis yang berjudul "Analisis Perbandingan Kinerja Algoritma Stemming Nazief-Adriani dan IN-Indris dalam Pengolahan Teks Bahasa Indonesia."

Tesis ini disusun sebagai salah satu syarat untuk memperoleh gelar Magister di Program PJJ S2 Teknik Informatika Universitas Amikom Yogyakarta. Penulis menyadari bahwa tanpa bimbingan, dukungan, dan doa dari berbagai pihak, penyusunan tesis ini tidak akan berjalan dengan lancar. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom., selaku pembimbing utama, dan Pak Anggit Dwi Hartanto, M.Kom., selaku pembimbing pendamping, yang telah dengan sabar memberikan arahan, masukan, dan motivasi dalam setiap tahap penyusunan tesis ini.
2. Seluruh dosen dan staf pengajar di Program PJJ S2 Teknik Informatika Universitas Amikom Yogyakarta atas ilmu dan bimbingan yang sangat berharga selama masa studi berlangsung.
3. Kedua orang tua tercinta, atas doa, cinta, dan dukungan yang tiada henti, baik secara moril maupun materiil.
4. Teman-teman seperjuangan, atas semangat, diskusi, dan kebersamaan yang sangat membantu dalam proses ini.

5. Semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu dan memberikan dukungan secara langsung maupun tidak langsung.

Penulis menyadari bahwa tesis ini masih jauh dari sempurna. Oleh karena itu, segala bentuk kritik dan saran yang membangun sangat penulis harapkan untuk perbaikan di masa mendatang. Akhir kata, penulis berharap semoga tesis ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang pemrosesan bahasa alami (Natural Language Processing) dan algoritma pemrosesan teks bahasa Indonesia.

Yogyakarta, 07 Agustus 2025

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

Penelitian ini bertujuan untuk mengevaluasi kinerja algoritma *stemming* Nazief-Adriani dan IN-Idris dalam pengolahan teks bahasa Indonesia, menganalisis pola kesalahan yang meliputi *over-stemming* dan *under-stemming*, serta mengusulkan perbaikan melalui pengembangan algoritma Hybrid Stemmer. Variabel yang diukur mencakup akurasi stemming dan waktu pemrosesan. Pengujian dilakukan pada tiga korpus teks formal berbahasa Indonesia, yaitu Preservasi Bahasa (1.888 kata), Sejarah NLP (2.733 kata), dan UUD 1945 (2.320 kata).

Metode penelitian yang digunakan adalah pendekatan kuantitatif eksperimental dengan membandingkan hasil *stemming* terhadap kata dasar yang telah dianotasi secara manual. Hasil penelitian menunjukkan bahwa algoritma IN-Idris menghasilkan akurasi lebih tinggi (82,46%–86,86%) dibandingkan algoritma Nazief-Adriani (74,35%–82,17%). Namun, kedua algoritma masih memiliki kelemahan dalam menangani kata berimbuhan kompleks, reduplikasi bermakna, dan istilah majemuk.

Algoritma Hybrid Stemmer yang dikembangkan dengan menggabungkan aturan dari kedua algoritma, serta penambahan exception list, normalisasi fonologi, dan penghapusan afiks ganda, mampu mencapai akurasi tertinggi sebesar 93,85% pada korpus Sejarah NLP, dengan peningkatan rata-rata lebih dari 10% dibandingkan kedua algoritma dasar. Pendekatan hybrid terbukti lebih efektif dalam pemrosesan teks formal berbahasa Indonesia, meskipun dengan konsekuensi peningkatan waktu pemrosesan. Penelitian selanjutnya disarankan untuk menguji algoritma pada teks informal, seperti media sosial, serta mengoptimalkan waktu eksekusi menggunakan teknik parallel processing atau struktur data Trie agar lebih efisien.

Kata kunci: *Stemming*, Hybrid Stemmer, NLP, Bahasa Indonesia, Morfologi

ABSTRACT

This study aims to evaluate the performance of the Nazief-Adriani and IN-Idris stemming algorithms in Indonesian text processing, analyze error patterns including over-stemming and under-stemming, and propose improvements through the development of a Hybrid Stemmer algorithm. The variables measured include stemming accuracy and processing time. The evaluation was conducted on three Indonesian formal text corpora: Preservasi Bahasa (1,888 words), Sejarah NLP (2,733 words), and UUD 1945 (2,320 words).

The research method employed a quantitative experimental approach by comparing stemming results with manually annotated root words. The results indicate that the IN-Idris algorithm achieves higher accuracy (82.46%–86.86%) compared to the Nazief-Adriani algorithm (74.35%–82.17%). However, both algorithms still exhibit limitations in handling complex affixed words, meaningful reduplication, and compound terms.

The proposed Hybrid Stemmer, which integrates the rules of both algorithms along with the addition of an exception list, phonological normalization, and multiple affix removal, achieves the highest accuracy of 93.85% on the Sejarah NLP corpus, with an average improvement of more than 10% over both baseline algorithms. The hybrid approach is proven to be more effective for processing formal Indonesian texts, albeit with increased processing time. Future research is recommended to evaluate the algorithm on informal texts such as social media and to optimize execution time using parallel processing techniques or Trie data structures for improved efficiency.

Keyword: *Stemming, Hybrid Stemmer, NLP, Indonesian Language, Morphology*

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Perkembangan teknologi informasi mendorong semakin meluasnya penggunaan teks dalam kehidupan sehari-hari, khususnya pada sektor pemerintahan, pendidikan, dan layanan publik. Di Indonesia, dokumen berbahasa formal seperti Undang-Undang, peraturan pemerintah, artikel ilmiah, dan buku panduan pendidikan menjadi bagian penting dari arus informasi nasional. Digitalisasi dokumen formal mendorong kebutuhan akan sistem pengolahan bahasa alami (Natural Language Processing/NLP) yang dapat memahami, mengolah, dan mengekstrak informasi dari teks berbahasa Indonesia dengan akurat (Amien, 2023).

Salah satu komponen utama dalam NLP adalah *stemming*, yaitu proses mengubah kata turunan menjadi kata dasar. Dalam konteks Bahasa Indonesia, proses ini menjadi lebih kompleks dibandingkan bahasa lain seperti Inggris karena struktur morfologi yang lebih kompleks. Bahasa Indonesia memiliki sistem afiksasi yang kaya, meliputi prefiks (me-, ber-, di-), sufiks (-kan, -i), konfiks (ke-an, per-an), hingga reduplikasi. Jika proses *stemming* dilakukan dengan cara yang tidak tepat, maka dapat menimbulkan kesalahan seperti *over-stemming* (pemotongan berlebihan hingga menghilangkan makna kata) atau *under-stemming* (pemotongan yang kurang sempurna sehingga kata masih mengandung imbuhan (Carolina et al., 2024).

Meskipun *Large Language Models* (LLM) seperti GPT-4 dan BERT telah membawa kemampuan pemahaman konteks yang tinggi, banyak sistem NLP masih bergantung pada tahap praproses berbasis aturan, termasuk *stemming*. Ini karena LLM memerlukan sumber daya komputasi yang besar dan belum sepenuhnya dioptimalkan untuk menangani struktur morfologi kompleks Bahasa Indonesia. Misalnya, bentuk reduplikasi seperti "mata-mata" atau variasi turunan seperti "mempersiapkan", "persiapan", dan "menyiapkan" tidak selalu dimaknai seragam oleh model, tanpa proses penyederhanaan eksplisit di awal (Nugraha, 2025). Selain keterbatasan morfologis, LLM juga memiliki bias terhadap bahasa mayoritas yang digunakan dalam data pelatihan—yang sebagian besar adalah bahasa Inggris. Oleh karena itu, dalam konteks Bahasa Indonesia, khususnya pada dokumen hukum, administrasi, atau edukasi yang membutuhkan interpretasi presisi, penggunaan *stemming* tetap menjadi pendekatan krusial untuk memastikan keseragaman representasi kata-kata penting (Nugraha, 2024). Penerapan *stemming* juga terbukti sangat berguna dalam sistem yang dirancang untuk sektor publik dan pemerintahan. Misalnya, dalam sistem pencarian arsip hukum dan dokumen kebijakan, *stemming* membantu mencocokkan berbagai bentuk kata turunan ke satu makna inti, sehingga meningkatkan hasil pencarian dan memperkecil kemungkinan informasi penting terlewat. Kasus seperti “mengatur”, “pengaturan”, dan “peraturan” semuanya dapat dikenali sebagai bentuk dasar “atur”, memperkuat akurasi pencarian teks di berbagai instansi pemerintahan (Jabbar et al., 2023). Hal serupa juga diterapkan pada sistem chatbot pelayanan publik, seperti di sektor perbankan atau lembaga pelayanan sosial. Dalam skenario ini, input dari pengguna sering kali tidak baku,

penuh variasi kata, dan bergantung pada konteks lokal. Dengan menerapkan *stemming* di tahap awal, sistem dapat menyederhanakan input dan mencocokkannya dengan basis pengetahuan atau daftar intent yang telah distandarisasi. Pendekatan ini membantu menstabilkan performa sistem NLP ringan yang dijalankan di infrastruktur terbatas (Mohemad et al., 2023). *Stemming* juga berkontribusi signifikan dalam sistem NLP berbasis pembelajaran mesin tradisional, seperti TF-IDF, Naïve Bayes, dan Support Vector Machine (SVM). Sistem ini masih digunakan secara luas untuk analisis teks berskala besar yang tidak memerlukan pemrosesan kontekstual kompleks. Tanpa *stemming*, model dapat mengalami masalah sparsitas karena terlalu banyak fitur unik akibat bentuk turunan yang bervariasi. Dengan *stemming*, sistem dapat mengurangi kompleksitas vektor dan mempercepat proses analisis (Rizki et al., 2023). Meskipun teknologi NLP telah berkembang pesat menuju pendekatan berbasis konteks dan model besar, teknik *stemming* tetap relevan dan kritikal, terutama dalam konteks Bahasa Indonesia yang memiliki struktur morfologi rumit. Penggunaan *stemming* mendukung efisiensi, akurasi, dan konsistensi dalam berbagai aplikasi NLP, mulai dari sistem hukum dan pemerintahan, layanan publik, hingga sistem edukasi dan komputasi ringan. Dalam konteks inilah, penelitian ini mengambil posisi strategis untuk mengevaluasi dan mengembangkan kembali metode *stemming* yang sesuai dengan kebutuhan lokal dan teknologi masa kini.

Di Indonesia, algoritma *stemming* yang paling populer digunakan adalah Nazief-Adriani, karena dianggap mampu menangani struktur morfologi Bahasa Indonesia secara komprehensif. Algoritma ini berbasis aturan (rule-based) yang

memanfaatkan kamus sebagai acuan validasi kata dasar. Namun, penelitian sebelumnya menunjukkan bahwa Nazief-Adriani masih memiliki beberapa keterbatasan, terutama saat dihadapkan pada teks dengan afiksasi ganda atau struktur kalimat yang lebih kompleks, seperti pada teks hukum dan peraturan perundang-undangan (Mustikasari et al., 2021).

Selain Nazief-Adriani, algoritma In-Idris hadir sebagai alternatif dengan pendekatan yang lebih sederhana dan cepat. In-Idris dikembangkan dari modifikasi algoritma Idris yang semula digunakan untuk Bahasa Melayu dan diadaptasi ke Bahasa Indonesia. Algoritma ini tidak bergantung pada kamus, melainkan murni berbasis aturan penghapusan imbuhan. Meskipun lebih ringan dari sisi pemrosesan, In-Idris memerlukan evaluasi lebih lanjut untuk membuktikan keakuratannya, terutama ketika digunakan pada teks formal yang memiliki standar kebahasaan lebih tinggi (Suci et al., 2022).

Hingga saat ini, sebagian besar penelitian terkait algoritma *stemming* di Indonesia masih berfokus pada teks populer seperti artikel berita atau data media sosial. Penelitian yang secara khusus menguji kinerja *stemming* pada korpus formal seperti dokumen hukum, artikel ilmiah, dan publikasi pemerintah masih terbatas (Rizki et al., 2023). Padahal, sektor formal memiliki kebutuhan yang lebih tinggi terhadap akurasi karena kesalahan dalam pemrosesan kata dapat berimplikasi pada makna yang keliru, terutama dalam konteks hukum dan administrasi. Selain itu, proses evaluasi algoritma *stemming* selama ini masih dilakukan secara manual menggunakan spreadsheet atau pencatatan konvensional. Metode tersebut berisiko menimbulkan inkonsistensi, sulit didokumentasikan, dan menyulitkan proses

kolaborasi antar peneliti atau anotator (Jabbar et al., 2023). Pengujian yang hanya mencatat akurasi numerik tanpa dokumentasi detail tentang jenis kesalahan seperti *over-stemming* dan *under-stemming* juga menghambat proses perbaikan algoritma di masa depan.

Untuk mendukung proses analisis dan perbaikan algoritma *stemming*, penelitian ini mengembangkan sebuah sistem evaluasi berbasis web yang berfungsi sebagai alat bantu validasi dan dokumentasi kesalahan secara sistematis. Sistem ini bukanlah fokus utama dari penelitian, melainkan sebagai sarana untuk memudahkan anotator dalam mengevaluasi hasil *stemming* secara lebih terstruktur dan transparan. Melalui sistem ini, anotator dapat:

- Melihat hasil *stemming* dari kedua algoritma (Nazief-Adriani dan In-Idris) secara langsung melalui antarmuka interaktif.
- Memberikan anotasi dan koreksi secara *real-time*, dengan mencatat apakah hasilnya sesuai dengan kaidah linguistik atau tidak.
- Mendokumentasikan jenis kesalahan yang terjadi, baik berupa *over-stemming*, *under-stemming*, maupun kesalahan lain yang relevan.
- Menyimpan riwayat validasi secara otomatis, sehingga proses evaluasi menjadi lebih transparan, reproducible, dan dapat diakses ulang oleh peneliti lain.

Penggunaan sistem anotasi dan validasi berbasis web seperti ini telah menjadi praktik yang direkomendasikan dalam NLP modern, karena dapat meningkatkan efisiensi serta akurasi dalam proses anotasi manual (Patil, 2022).

Sistem evaluasi berbasis web yang dikembangkan dalam penelitian ini berfungsi sebagai alat bantu untuk mendukung proses validasi dan dokumentasi kesalahan secara lebih efisien dan transparan. Dengan sistem ini, proses anotasi menjadi lebih terstruktur sehingga memudahkan analisis jenis kesalahan yang muncul, seperti *over-stemming* dan *under-stemming*. Fokus utama dari penelitian ini adalah mengevaluasi performa kedua algoritma *stemming* yang akan dibandingkan serta memberikan saran perbaikan terhadap kelemahan yang ditemukan. Hasil evaluasi ini diharapkan dapat menjadi dasar untuk mengusulkan pengembangan aturan baru atau penyesuaian metode yang lebih efektif dalam pengolahan teks berbahasa Indonesia di sektor formal seperti pemerintahan, pendidikan, dan hukum.

1.2. Rumusan Masalah

Algoritma Nazief-Adriani telah lama menjadi standar dalam proses *stemming* untuk Bahasa Indonesia dan sudah banyak diteliti performanya dalam berbagai konteks. Sebaliknya, algoritma In-Idris, yang merupakan adaptasi dari metode Idris untuk Bahasa Melayu, masih sangat jarang dievaluasi secara komprehensif dalam pemrosesan Bahasa Indonesia.

Sampai saat ini, belum ada penelitian yang secara langsung membandingkan performa Nazief-Adriani dengan In-Idris. Padahal, perbandingan ini penting untuk mengetahui apakah algoritma yang lebih sederhana seperti In-Idris dapat menjadi alternatif yang layak atau memerlukan penyesuaian lebih lanjut untuk mencapai hasil yang setara atau lebih baik.

Selain itu, penelitian sebelumnya cenderung hanya membahas perbandingan akurasi secara numerik tanpa mendalami pola kesalahan yang muncul. Dokumentasi kesalahan seperti *over-stemming* dan *under-stemming* sangat penting untuk menemukan titik kelemahan algoritma secara lebih rinci, agar perbaikan yang diusulkan benar-benar relevan dengan kebutuhan praktis.. Berdasarkan kondisi tersebut, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana performa algoritma Nazief-Adriani dan In-Idris dalam proses *stemming* terhadap korpus Bahasa Indonesia?
2. Apa saja kelemahan dan pola kesalahan yang ditemukan dari hasil evaluasi kedua algoritma tersebut, khususnya terkait over-stemming dan under-*stemming*?
3. Bagaimana rekomendasi atau saran perbaikan terhadap algoritma *stemming* berbasis aturan agar lebih optimal untuk pemrosesan Bahasa Indonesia?

1.3. Batasan Masalah

Agar penelitian ini memiliki ruang lingkup yang jelas dan terfokus, perlu ditetapkan beberapa batasan agar tujuan yang ingin dicapai dapat dilakukan secara optimal. Penetapan batasan ini penting untuk menghindari pelebaran kajian yang tidak relevan dengan tujuan utama, yaitu evaluasi performa algoritma dan pemberian saran perbaikan untuk algoritma *stemming* berbasis aturan. Batasan dalam penelitian ini adalah sebagai berikut:

1. Data yang digunakan dalam penelitian ini terbatas pada korpus berbahasa Indonesia yang bersifat baku dan sesuai dengan kaidah ejaan.

2. Penelitian hanya membandingkan dua algoritma *stemming* berbasis aturan, yaitu Nazief-Adriani dan In-Idris.
3. Perbaikan yang diusulkan dalam penelitian ini terbatas pada saran pengembangan atau penyesuaian aturan di dalam algoritma berbasis *rule-based*.

1.4. Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan untuk memberikan kontribusi terhadap pengembangan algoritma *stemming* berbasis aturan dalam Bahasa Indonesia, khususnya dalam meningkatkan keakurasian dan ketepatan pemotongan kata ke bentuk dasarnya. Selain itu, penelitian ini bertujuan untuk menghasilkan evaluasi yang lebih rinci, tidak hanya dari sisi akurasi secara umum tetapi juga dari pola kesalahan yang terjadi. Adapun tujuan khusus dari penelitian ini adalah sebagai berikut:

1. Mengevaluasi performa algoritma Nazief-Adriani dan In-Idris dalam proses *stemming* terhadap korpus Bahasa Indonesia yang bersifat baku.
2. Menganalisis kelemahan kedua algoritma tersebut dengan mendokumentasikan jenis kesalahan yang terjadi, terutama *over-stemming* dan *under-stemming*.
3. Memberikan saran perbaikan atau rekomendasi pengembangan aturan pada algoritma *stemming* berbasis aturan agar lebih optimal dalam memproses Bahasa Indonesia.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat baik secara teoritis maupun praktis, khususnya dalam pengembangan teknologi pemrosesan bahasa alami untuk Bahasa Indonesia. Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Manfaat Teoritis
    - Menambah kajian ilmiah di bidang pemrosesan bahasa alami, khususnya terkait evaluasi dan pengembangan algoritma *stemming* berbasis aturan untuk Bahasa Indonesia.
    - Menyediakan dokumentasi pola kesalahan seperti *over-stemming* dan *under-stemming* sebagai referensi bagi peneliti lain yang ingin mengembangkan atau menyempurnakan algoritma serupa.
    - Memberikan referensi perbandingan kinerja antara algoritma Nazief-Adriani dan In-Idris yang selama ini belum pernah dilakukan secara langsung.
2. Manfaat Praktis
    - Memberikan rekomendasi atau saran perbaikan pada aturan algoritma *stemming* agar dapat diimplementasikan dengan lebih baik di berbagai aplikasi pengolahan teks Bahasa Indonesia, seperti klasifikasi dokumen, pencarian informasi, dan pengindeksan dokumen hukum atau akademik.

- Menyediakan data hasil evaluasi yang lebih rinci dan sistematis, yang dapat digunakan oleh pengembang sistem NLP untuk menyesuaikan algoritma dengan kebutuhan praktis di lapangan.
- Memudahkan proses validasi hasil *stemming* melalui sistem evaluasi berbasis web yang dapat digunakan sebagai alat bantu dalam dokumentasi kesalahan dan proses anotasi secara terstruktur.

3. Manfaat Sosial dan Budaya
   - Mendukung pelestarian dan pemrosesan Bahasa Indonesia secara lebih akurat dalam ranah digital, sehingga meningkatkan kualitas aplikasi berbasis bahasa dalam sektor pendidikan, pemerintahan, dan layanan publik.
   - Membantu memperkuat ekosistem teknologi berbahasa Indonesia dengan menyediakan algoritma yang lebih sesuai dengan kaidah dan struktur bahasa yang baku, sekaligus mendorong kemandirian teknologi di bidang pemrosesan bahasa lokal.
   - Memfasilitasi pengembangan sistem informasi berbasis teks yang dapat menjaga konsistensi bahasa sesuai dengan aturan yang berlaku, sehingga turut menjaga ketertiban penggunaan bahasa di lingkungan sosial dan budaya.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian mengenai *stemming* dalam bahasa Indonesia telah berkembang pesat seiring meningkatnya kebutuhan akan sistem pemrosesan teks yang akurat dan efisien. Kompleksitas morfologi bahasa Indonesia, yang kaya akan prefiks, sufiks, infiks, dan konfiks, menuntut pendekatan algoritmik yang mampu mengembalikan kata ke bentuk dasarnya secara tepat. Oleh karena itu, berbagai algoritma *stemming* telah dikembangkan dan diuji dalam berbagai konteks NLP.

Penelitian oleh (Mohemad et al., 2023) menjadi salah satu studi penting yang membandingkan algoritma *stemming* berbasis aturan untuk bahasa Melayu, yaitu algoritma Othman dan Ahmad. Kajian ini meneliti bagaimana algoritma tersebut beroperasi pada dua domain dataset yang berbeda: ringkasan kursus dan laporan kejahatan. Algoritma Ahmad, yang menggunakan 561 aturan morfologi, menunjukkan akurasi tertinggi, mencapai 93,80%, dibandingkan algoritma Othman. Penelitian ini memberikan wawasan bahwa algoritma berbasis aturan dapat memberikan kinerja yang sangat baik jika didukung oleh pemahaman morfologi bahasa yang mendalam (Mohemad et al., 2023).

Untuk bahasa Indonesia, algoritma *stemming* yang sering digunakan adalah Nazief-Adriani dan Porter. (Tuhpatussania et al., 2022) melakukan studi komparatif terhadap kedua algoritma tersebut, dengan fokus pada pengolahan teks modul pembelajaran bahasa Indonesia. Nazief-Adriani menunjukkan akurasi yang lebih

tinggi dibandingkan Porter, masing-masing sebesar 74,175% dan 73,225%. Namun, Nazief-Adriani membutuhkan waktu proses yang lebih lama karena kompleksitas aturan imbuhan yang diterapkan. Penelitian ini menunjukkan bahwa pendekatan berbasis aturan dapat memberikan hasil yang baik, tetapi efisiensinya menjadi tantangan, terutama ketika diterapkan pada dataset besar (Tuhpatussania et al., 2022).

(Suci et al., 2022) mengembangkan algoritma In-Idris sebagai bentuk modifikasi dari Idris yang disesuaikan untuk bahasa Indonesia. Penelitian ini bertujuan untuk meningkatkan akurasi serta efisiensi dalam pemrosesan teks bahasa Indonesia. Hasil yang diperoleh menunjukkan bahwa algoritma In-Idris mencapai akurasi 82,81%, mengalami peningkatan sebesar 5,25% dibandingkan dengan versi asli Idris, serta memiliki kinerja pemrosesan yang lebih cepat. Temuan ini menegaskan bahwa pengoptimalan algoritma *stemming* yang disesuaikan dengan struktur morfologi bahasa Indonesia dapat meningkatkan kinerja pemrosesan bahasa alami (NLP) (Suci et al., 2022).

(Mustikasari et al., 2021) melakukan penelitian untuk membandingkan efektivitas berbagai algoritma *stemming* dalam pemrosesan dokumen berbahasa Indonesia. Hasil studi menunjukkan bahwa algoritma Nazief-Adriani memiliki akurasi sebesar 92,4%, sedangkan algoritma Sastrawi mencapai tingkat akurasi tertinggi, yaitu 95,2%. Temuan ini mengindikasikan bahwa meskipun Nazief-Adriani cukup andal, Sastrawi sebagai algoritma yang lebih modern memiliki keunggulan dalam mengenali kata dasar dengan lebih baik, terutama saat diterapkan pada teks yang lebih kompleks (Mustikasari et al., 2021).

Penelitian oleh (Jumadi et al., 2021) membandingkan algoritma *stemming* Nazief-Adriani dan Paice-Husk dalam pemrosesan teks bahasa Indonesia. Hasil penelitian menunjukkan bahwa Nazief-Adriani memiliki akurasi lebih tinggi (91,87%) dibandingkan Paice-Husk (64,43%), meskipun waktu pemrosesan Nazief-Adriani lebih lama. Studi ini mengonfirmasi bahwa pendekatan berbasis aturan lebih efektif dalam bahasa Indonesia dibandingkan algoritma Paice-Husk yang lebih umum digunakan dalam bahasa Inggris (Jumadi et al., 2021).

(Jabbar et al., 2023) mengeksplorasi berbagai teknik *stemming* dalam konteks NLP dan Information Retrieval (IR). Hasil analisis mereka mengungkapkan bahwa metode yang berlandaskan linguistik cenderung lebih akurat ketika diterapkan pada bahasa dengan struktur morfologi yang kompleks. Meskipun demikian, pendekatan berbasis aturan memiliki kelemahan, terutama dalam hal konsumsi sumber daya pemrosesan yang lebih tinggi, sehingga kurang efisien untuk menangani data dalam skala besar (Jabbar et al., 2023).

Penelitian oleh (Rizki et al., 2023) meneliti implementasi algoritma Nazief-Adriani dalam pemrosesan data Twitter yang mengandung campuran bahasa Indonesia dan Banjar. Hasil studi menunjukkan bahwa algoritma ini memiliki akurasi sebesar 90,34%, membuktikan efektivitasnya dalam menangani teks sosial media. Studi ini menyoroti pentingnya optimalisasi *stemming* untuk bahasa campuran dalam konteks analisis media sosial (Rizki et al., 2023).

Penelitian-penelitian ini memberikan dasar yang kuat untuk penelitian ini, terlihat bahwa algoritma *stemming* untuk bahasa Indonesia terus mengalami perkembangan dengan berbagai pendekatan. Algoritma Nazief-Adriani masih

menjadi pilihan utama karena keakuratannya yang tinggi dalam menemukan kata dasar. Namun, penelitian juga menunjukkan bahwa terdapat alternatif lain seperti Sastrawi dan In-Idris yang menawarkan keunggulan dalam aspek tertentu, terutama dalam kecepatan dan efisiensi pemrosesan. Beberapa studi menyoroti bahwa pendekatan berbasis aturan lebih unggul dalam menangani kompleksitas morfologi bahasa Indonesia, tetapi memerlukan waktu pemrosesan yang lebih lama dibandingkan metode statistik seperti Porter dan Paice-Husk.

Studi ini menjadi dasar bagi penelitian lebih lanjut dalam membandingkan kinerja algoritma *stemming* Nazief-Adriani dan In-Idris dalam pemrosesan teks bahasa Indonesia. Dengan mengevaluasi akurasi, kecepatan, dan efisiensi kedua algoritma, penelitian ini diharapkan dapat memberikan wawasan yang lebih dalam mengenai algoritma *stemming* yang paling optimal untuk digunakan dalam berbagai aplikasi NLP dan analisis teks bahasa Indonesia.

2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian
Analisis Perbandingan Kinerja Algoritma *Stemming* Nazief-Adriani Dan In-Idris Dalam Pengolahan Teks Bahasa Indonesia

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	*Comparison of Porter's and Nazief-Adriani's Algorithm*	Tuhpatussania et al., Pilar Nusa Mandiri, 2022	Membandingkan algoritma Porter dan Nazief-Adriani dalam modul pembelajaran bahasa	Nazief-Adriani lebih akurat (74,175%)	Nazief-Adriani memiliki waktu proses lebih lama karena struktur aturan morfologinya kompleks	Penelitian ini menggunakan pendekatan komparatif terbatas antara algoritma klasik dan lokal. Sementara penelitian yang akan dilakukan menggunakan dua algoritma lokal (Nazief-Adriani & In-Idris) yang sama-sama berbasis aturan namun dengan pendekatan yang berbeda. Penelitian ini juga lebih kompleks karena mencakup data dari berbagai domain dan evaluasi efisiensi.
2	*IN-IDRIS: Modification of Idris for Indonesian Text*	Suci et al., IIUM Engineering Journal, 2022	Mengembangkan dan menguji algoritma In-Idris untuk bahasa Indonesia	Akurasi 82,81% dan lebih cepat dari Idris	Perlu uji lanjutan dengan dataset lebih besar dan perbandingan eksplisit	Penelitian ini hanya fokus pada pengembangan dan pengujian In-Idris secara mandiri. Penelitian yang akan dilakukan mengevaluasi kinerja In-Idris secara langsung terhadap Nazief-Adriani dengan metode eksperimental terstruktur dan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
						parameter evaluasi yang jelas (akurasi dan waktu).
3	*Comparison of Nazief-Adriani and Paice-Husk Algorithm*	Jumadi et al., IOP Conf. Series, 2021	Membandingkan performa dua algoritma untuk teks Indonesia	Nazief-Adriani lebih akurat (91,87%)	Paice-Husk tidak cocok untuk struktur bahasa Indonesia	Perbedaan utama terletak pada kedalaman evaluasi. Penelitian sebelumnya membandingkan algoritma yang berbeda karakter (Indonesia vs Inggris). Penelitian yang akan dilakukan membandingkan dua algoritma yang sama-sama disesuaikan dengan struktur bahasa Indonesia untuk validasi lebih kontekstual.
4	*Effectiveness of Stemming Algorithms in Indonesian Documents*	Mustikasari et al., Borobudur Symposium, 2021	Menilai efektivitas berbagai algoritma termasuk Sastrawi	Sastrawi lebih akurat (95,2%)	Sastrawi lambat, tidak cocok untuk semua kasus	Fokus penelitian tersebut adalah pembandingan luas antar banyak algoritma tanpa analisis khusus. Penelitian yang akan dilakukan hanya berfokus pada dua algoritma yang berbasis aturan dan digunakan secara nyata di berbagai aplikasi NLP, dengan skenario uji dan beban kerja nyata.

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	*Implementation of Stemming on Twitter Data*	Rizki et al., Fidelity Journal, 2023	Menguji Nazief-Adriani untuk teks sosial media	Nazief-Adriani cukup efektif (90,34%)	Kesulitan menangani kata tidak baku	Penelitian ini terbatas pada satu algoritma dan satu jenis data (Twitter). Penelitian yang akan dilakukan lebih kompleks karena membandingkan dua algoritma serta menguji kinerja pada tiga domain data (berita, akademik, media sosial) yang memiliki kompleksitas linguistik yang berbeda-beda.
6	*Comparative Study of Stemming Techniques on Malay Text*	Mohemad et al., IJACSA, 2023	Membandingkan algoritma untuk bahasa Melayu	Algoritma Ahmad lebih akurat (93,8%)	Studi hanya menggunakan bahasa Melayu	Penelitian ini berbeda secara fundamental dari segi bahasa, struktur linguistik, dan tujuan. Fokus penelitian yang akan dilakukan lebih inovatif karena menguji adaptasi lintas algoritma Melayu ke Indonesia (In-Idris), serta membandingkannya dengan algoritma lokal Indonesia.
7	*Text Stemming Methodologies in NLP and IR*	Jabbar et al., IEEE Access, 2023	Mengkaji pendekatan *stemming* dalam NLP/IR	Linguistik-based lebih cocok untuk bahasa kompleks	Aturan linguistik cenderung boros komputasi	Penelitian ini bersifat survei metodologis secara umum. Penelitian yang akan dilakukan bersifat empiris-komparatif dengan eksperimen aktual, mencakup

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
						pengukuran akurasi dan waktu, sehingga menghasilkan kontribusi praktis dan terukur.

2.3. Landasan Teori

Landasan teori memberikan dasar konseptual yang menjadi pijakan dalam pelaksanaan dan analisis penelitian. Dalam konteks penelitian ini, teori-teori yang relevan mencakup bidang Natural Language Processing (NLP), teknik *stemming*, serta penjabaran algoritma Nazief-Adriani dan In-Idris sebagai metode utama yang dibandingkan.

a) *Natural Language Processing* (NLP)

NLP adalah cabang kecerdasan buatan yang berfokus pada interaksi antara komputer dan manusia melalui bahasa alami. NLP melibatkan berbagai teknik untuk memahami, menganalisis, dan menghasilkan teks atau ucapan manusia. Beberapa aplikasi utama NLP meliputi pencarian informasi, text mining, analisis sentimen, klasifikasi teks, serta sistem tanya jawab. Dalam konteks bahasa dengan struktur morfologi kompleks seperti bahasa Indonesia, salah satu tantangan utama dalam NLP adalah mengelola variasi morfologi untuk meningkatkan akurasi pemrosesan teks. Salah satu teknik penting untuk menangani tantangan ini adalah *stemming*, yang mengembalikan kata berimbuhan ke bentuk dasar atau kata dasarnya (Geetha et al., 2023).

b) *Stemming*

Stemming merupakan salah satu tahapan penting dalam proses pra-pemrosesan teks dalam *Natural Language Processing* (NLP). Tujuan utama dari *stemming* adalah untuk mengurangi variasi kata menjadi bentuk dasarnya (root word), sehingga sistem komputer dapat mengelompokkan dan menganalisis teks

secara lebih efektif. Contohnya, kata-kata seperti "bermain," "memainkan," dan "mainan" akan dikembalikan ke bentuk dasar "main."

Dalam bahasa Indonesia, proses *stemming* menjadi lebih menantang dibandingkan bahasa seperti Inggris karena adanya struktur morfologi yang kompleks, seperti penggunaan prefiks (me-, di-, ber-), sufiks (-kan, -i), maupun konfiks (ke-an, memper-an). Jika tidak ditangani secara tepat, *stemming* dapat menimbulkan kesalahan seperti *over-stemming* (kata yang terlalu dipotong) atau *under-stemming* (imbuhan tidak berhasil dihapus sepenuhnya), yang dapat menurunkan kualitas analisis data teks secara keseluruhan (Carolina et al., 2024).

c) Algoritma Nazief-Adriani

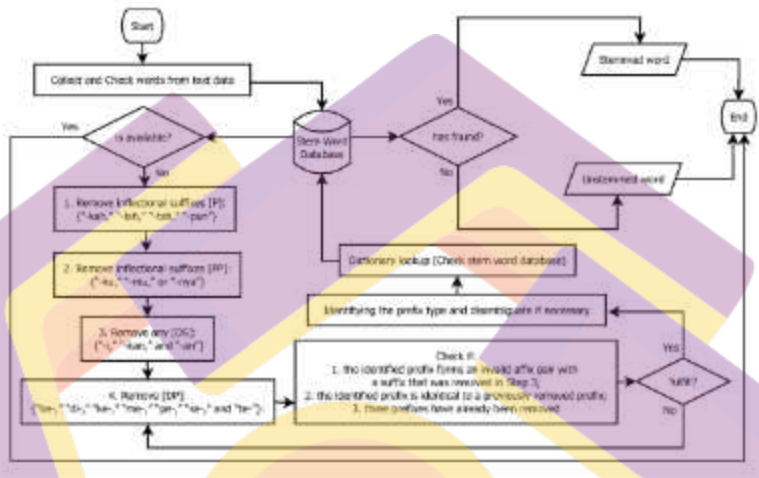
Algoritma Nazief-Adriani merupakan salah satu algoritma stemming berbasis kamus yang banyak digunakan dalam pemrosesan bahasa alami (Natural Language Processing). Algoritma ini dikembangkan oleh Nazief dan Adriani untuk menangani morfologi bahasa Indonesia yang kompleks, terutama dalam hal imbuhan seperti prefiks (awalan), sufiks (akhiran), infiks (sisipan), dan konfiks (kombinasi awalan dan akhiran).

Langkah – langkah algoritma Nazief dan Adriani dalam pemrosesan teks adalah sebagai berikut:

1. **Pemeriksaan dalam Kamus**: Kata yang diinput akan diperiksa terlebih dahulu dalam kamus kata dasar. Jika kata tersebut sudah merupakan kata dasar, maka tidak dilakukan pemrosesan lebih lanjut.

2. **Pemisahan Imbuhan**: Jika kata tidak ditemukan dalam kamus, maka sistem akan mencoba menghapus imbuhan yang terdiri dari prefiks, sufiks, infiks, dan konfiks menggunakan aturan morfologi bahasa Indonesia.
3. **Pengecekan Validitas Kata Dasar**: Setelah imbuhan dihapus, kata yang dihasilkan akan dicek ulang dalam kamus untuk memastikan bahwa kata tersebut benar-benar merupakan kata dasar yang valid.: Setelah imbuhan dihapus, kata yang dihasilkan akan dicek ulang dalam kamus untuk memastikan bahwa kata tersebut benar-benar merupakan kata dasar yang valid.
4. **Penerapan Aturan Tambahan**: Jika kata yang dihasilkan tidak ditemukan dalam kamus, algoritma akan menerapkan aturan tambahan, seperti pemrosesan kata turunan atau pengecekan ulang untuk imbuhan tertentu yang mungkin salah dihapus.

Meskipun dikenal akurat, algoritma ini memiliki kelemahan dalam hal efisiensi waktu pemrosesan. Proses pencocokan ke dalam kamus dan penghapusan berlapis membutuhkan sumber daya komputasi lebih besar, terutama saat diaplikasikan pada dataset besar (Jumadi et al., 2021). Gambar 2.1 menunjukkan alur kerja dari proses *stemming* pada algoritma Nazief & Adriani.



Gambar 2.1 - Diagram Alur Nazief & Adriani

d) Algoritma In-Idris

In-Idris merupakan modifikasi dari algoritma Idris yang semula dikembangkan untuk bahasa Melayu. Algoritma ini kemudian disesuaikan dengan struktur morfologi bahasa Indonesia. Tidak seperti Nazief-Adriani, In-Idris mengandalkan pendekatan berbasis aturan sepenuhnya tanpa menggunakan kamus sebagai acuan. Oleh karena itu, algoritma ini lebih ringan dan cepat dalam memproses teks. Proses In-Idris berfokus pada penghapusan imbuhan melalui analisis pola huruf awal dan akhir berdasarkan aturan tertentu. Algoritma ini menerapkan urutan pemeriksaan terhadap prefiks dan sufiks, dan dalam beberapa kasus melakukan kombinasi analisis karakteristik fonologis kata.

Langkah-langkah dalam algoritma Idris meliputi:

1. Periksa kata dalam kamus. Jika kata tersebut ada dalam kamus, maka kata dianggap sebagai kata dasar dan keluar. Jika tidak, lanjut ke langkah berikutnya.
2. Periksa kata dalam aturan prefiks. Jika kata sesuai dengan aturan prefiks, maka hapus prefiks dan lanjut ke langkah 6. Jika tidak, lanjut ke langkah berikutnya.
3. Periksa pola prefiks dan huruf pertama dari kata yang akan di *stemming*. Jika pola prefiks cocok dengan Aturan 2, maka terapkan Aturan 2 pada kata tersebut. Jika tidak, hapus prefiks dan lanjut ke langkah 6.
4. Periksa prefiks kata dan sesuaikan pola dalam Aturan 2. Jika cocok dengan aturan keempat, maka periksa kata dasar dalam kamus dan lanjut ke langkah berikutnya. Jika tidak, hapus prefiks dan lanjut ke langkah 6.
5. Jika kata tidak ditemukan dalam kamus, maka kembali ke langkah 4. Jika ditemukan, hapus prefiks dan lanjut ke langkah berikutnya.
6. Periksa kata dalam kamus. Jika kata ditemukan, maka kata dianggap sebagai kata dasar dan keluar. Jika tidak, lanjut ke langkah berikutnya.
7. Periksa kata dalam aturan sufiks. Jika cocok dengan aturan sufiks, maka hapus sufiks dan kembali ke langkah 1. Jika tidak, langsung kembali ke langkah 1.

Pada Gambar 2.2 merupakan ilustrasi bagaimana cara kerja algoritma In-Idris dalam melakukan proses *stemming*.

Gambar 2.2 - Diagram Alur algoritma In-Idris

Keunggulan utama algoritma In-Idris terletak pada efisiensinya dalam memproses teks, sehingga cocok diterapkan dalam sistem dengan keterbatasan sumber daya atau kebutuhan *real-time*. Namun, karena tidak bergantung pada kamus, potensi kesalahan *stemming* lebih tinggi pada kata tidak umum atau bentuk yang ambigu (Suci et al., 2022).

e) Evaluasi Kinerja Algoritma

Evaluasi algoritma *stemming* dilakukan untuk mengukur seberapa baik algoritma mengembalikan kata ke bentuk dasarnya. Dua metrik utama yang digunakan adalah:

1. Akurasi

   Mengukur sejauh mana hasil stemming sesuai dengan kata dasar yang benar. Akurasi dihitung dari persentase kata yang berhasil diubah ke bentuk dasar yang valid dibandingkan dengan jumlah total kata dalam dataset.

2. Waktu Pemrosesan

   Mengukur efisiensi algoritma dari sisi kecepatan pemrosesan data. Metrik ini penting terutama untuk aplikasi real-time atau sistem yang harus memproses data dalam volume besar.

Selain itu, untuk mendalami kualitas hasil, analisis lanjutan dapat menggunakan confusion matrix, serta metrik precision, recall, dan F1-score sebagai indikator tambahan untuk mengevaluasi keseimbangan antara akurasi dan cakupan hasil *stemming* (Jabbar et al., 2023).

f) Teknologi Terkait

Beberapa algoritma *stemming* telah dikembangkan untuk bahasa Indonesia dan bahasa lain yang memiliki karakteristik morfologi yang kompleks. Berikut adalah beberapa teknologi terkait yang memiliki kesamaan dalam pemrosesan teks bahasa Indonesia.

1. Sastrawi

   Sastrawi adalah salah satu algoritma *stemming* berbasis aturan yang banyak digunakan untuk bahasa Indonesia. Algoritma ini dikembangkan berdasarkan Enhanced Confix Stripping (ECS) yang merupakan penyempurnaan dari metode *stemming* sebelumnya. Langkah – langkah dalam melakukan proses *stemming* algoritma ini adalah sebagai berikut :

   - **Tokenisasi**: Memisahkan teks menjadi kata-kata individual.
   - **Penerapan Aturan Penghapusan Imbuhan**: Menghapus prefiks, infiks, dan sufiks menggunakan aturan yang telah ditetapkan.
   - **Pengecekan Kata Dasar**: Memeriksa hasil *stemming* dengan daftar kata dasar yang valid.
   - **Validasi dan Koreksi**: Jika kata dasar tidak ditemukan, dilakukan langkah tambahan untuk menangani kasus pengecualian (Mustikasari et al., 2021).

2. Porter Stemmer

   Porter Stemmer adalah salah satu algoritma *stemming* yang paling banyak digunakan dalam bahasa Inggris dan telah diadaptasi untuk bahasa Indonesia (Arif Siswandi et al., 2021). Cara kerja dari algoritma ini adalah sebagai berikut :

   - **Identifikasi Akhiran Umum**: Algoritma menghapus sufiks yang umum dalam bahasa yang diproses.
   - **Transformasi Bentuk Kata**: Mengubah kata ke bentuk dasarnya berdasarkan aturan morfologi tertentu.

- **Pengecekan Validitas**: Jika kata hasil stemming tidak valid, proses dilakukan ulang dengan aturan lain (Arif Siswandi et al., 2021).

3. Paice-Husk Stemmer

Paice-Husk Stemmer adalah algoritma *stemming* berbasis aturan yang menggunakan daftar aturan pemotongan yang diurutkan berdasarkan prioritas. Langkah – Langkah pada pemrosesan algoritma ini adalah sebagai berikut:

- **Daftar Aturan Pemotongan**: Setiap kata diperiksa dan dipotong berdasarkan aturan yang tersedia.
- **Iterasi Pemrosesan**: Jika kata yang dihasilkan tidak valid, proses diulang dengan aturan lain hingga kata dasar ditemukan.
- **Validasi Akhir**: Jika tidak ada kata dasar yang ditemukan, kata asli dikembalikan sebagai hasil akhir (Jumadi et al., 2021).

g) Relevansi Penelitian

Penelitian ini memiliki relevansi yang tinggi dalam konteks akademik dan praktis, khususnya di bidang pemrosesan bahasa alami (Natural Language Processing/NLP) untuk bahasa Indonesia. Kompleksitas struktur morfologi bahasa Indonesia menuntut pengembangan algoritma *stemming* yang tidak hanya akurat, tetapi juga efisien dalam memproses teks dari berbagai domain. Secara akademik, penelitian ini memberikan kontribusi dalam bentuk:

- Pengayaan kajian komparatif algoritma lokal: Dengan membandingkan langsung algoritma Nazief-Adriani dan In-Idris dalam satu eksperimen

yang sama, penelitian ini mengisi celah dalam literatur yang sebelumnya hanya mengkaji masing-masing algoritma secara terpisah.

- Pengujian pada multi-domain data: Penelitian ini menggunakan data dari berbagai sumber (berita, akademik, dan media sosial), sehingga memberikan pemahaman yang lebih menyeluruh terhadap kinerja algoritma di lingkungan nyata.
- Analisis dual-metrik (akurasi dan efisiensi): Evaluasi tidak hanya dilakukan berdasarkan akurasi hasil *stemming*, tetapi juga efisiensi waktu, yang sering kali diabaikan dalam penelitian sebelumnya.

Secara praktis, relevansi penelitian ini juga tinggi karena:

- Mendukung pengembangan sistem NLP berbahasa Indonesia seperti chatbot, mesin pencari lokal, dan analisis sentimen media sosial, yang sangat bergantung pada hasil stemming yang akurat dan cepat.
- Membantu pengembang aplikasi untuk memilih metode *stemming* yang paling sesuai dengan kebutuhan spesifik, baik dalam konteks skala kecil (real-time) maupun besar (big data).
- Mendorong efisiensi pemrosesan teks di sektor industri dan pemerintahan, khususnya dalam klasifikasi dokumen, pelacakan opini publik, serta digitalisasi arsip dan dokumen resmi.

Dengan menggabungkan aspek linguistik, teknologi, dan kebutuhan aplikasi modern, penelitian ini diharapkan dapat memberikan kontribusi nyata terhadap pengembangan teknologi berbasis bahasa Indonesia, sekaligus memperkaya kajian ilmiah di bidang NLP lokal yang masih relatif terbatas.

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini tergolong dalam jenis penelitian kuantitatif eksperimental, yang bertujuan untuk melakukan pengujian dan evaluasi langsung terhadap dua algoritma *stemming* Nazief-Adriani dan In-Idris dengan parameter terukur seperti akurasi dan waktu proses. Pendekatan ini memungkinkan peneliti memperoleh bukti empiris secara objektif terkait kinerja masing-masing algoritma dalam pemrosesan teks bahasa Indonesia. Sifat penelitian ini adalah komparatif-aplikatif, karena membandingkan performa dua algoritma *rule-based* yang umum digunakan dalam bidang NLP untuk bahasa Indonesia. Penelitian tidak hanya bersifat pengujian teoretis, tetapi juga diarahkan pada implementasi nyata yang dapat digunakan oleh pengembang sistem informasi, peneliti NLP, maupun praktisi teknologi teks. Dalam studi-studi sebelumnya, pendekatan komparatif juga digunakan untuk mengevaluasi keunggulan dan kelemahan berbagai algoritma stemming seperti Porter, Nazief-Adriani, Paice-Husk, hingga modifikasi seperti In-Idris dan ECS (Jumadi et al., 2021), (Carolina et al., 2024). Penelitian ini memperluas pendekatan tersebut dengan menerapkan evaluasi ganda (akurasi dan waktu) dan pengujian lintas domain.

Dalam rangka mendukung proses evaluasi yang efisien dan terstruktur, penelitian ini juga memanfaatkan sebuah *supporting tool* berupa platform berbasis web. Platform ini dikembangkan secara khusus untuk memfasilitasi validasi hasil

stemming yang dilakukan oleh dua algoritma, yakni Nazief-Adriani dan In-Idris. Sistem ini memungkinkan proses evaluasi dilakukan secara semi-otomatis, menggabungkan validasi manual oleh pengguna melalui antarmuka interaktif dan juga validasi otomatis menggunakan alat *lemmatization* seperti Stanza. Dengan pendekatan ini, peneliti dapat melakukan analisis akurasi dan efisiensi secara sistematis terhadap korpus yang besar dan heterogen. Perlu ditegaskan bahwa pengembangan sistem tersebut tidak menjadi objek utama penelitian ini, melainkan berfungsi sebagai media bantu untuk memperlancar dan memperkuat proses pengumpulan serta analisis data hasil *stemming*.

3.2. Metode Pengumpulan Data

Dalam penelitian ini, proses pengujian algoritma dilakukan dengan menggunakan *corpus* yang disusun secara mandiri. Hal ini didasarkan pada kenyataan bahwa sebagian besar penelitian terdahulu mengenai algoritma stemming, baik Nazief-Adriani maupun In-Idris, tidak menyertakan dataset secara eksplisit. Kondisi tersebut menyebabkan replikasi langsung terhadap hasil penelitian terdahulu sulit dilakukan. Oleh karena itu, diperlukan penyusunan *corpus* baru agar perbandingan algoritma dapat dilakukan secara objektif dan terkontrol.

Corpus yang digunakan dalam penelitian ini terdiri atas kumpulan teks berbahasa Indonesia yang bersumber dari berita daring, abstrak tugas akhir, serta dokumen pendidikan. Pemilihan *corpus* dilakukan dengan tujuan untuk merepresentasikan variasi morfologi yang umum ditemukan dalam bahasa Indonesia, sehingga kinerja algoritma dapat diuji secara lebih menyeluruh. Dengan

adanya *corpus* ini, seluruh algoritma yang dibandingkan—yaitu Nazief-Adriani, In-Idris, serta algoritma hybrid yang diusulkan—dijalankan pada dataset yang sama, sehingga hasil evaluasi menjadi adil dan dapat diandalkan. Pendekatan ini memastikan bahwa perbedaan performa yang muncul lebih tepat dikaitkan dengan karakteristik algoritma, bukan dengan variasi dataset. Dengan demikian, hasil pengujian dapat merefleksikan kelebihan dan kelemahan masing-masing metode secara lebih akurat. Objek penelitian dalam studi ini adalah hasil *stemming* dari dua algoritma berbasis aturan, yaitu Nazief-Adriani dan In-Idris, yang diuji pada *corpus* berbahasa Indonesia. Data yang digunakan berupa teks berbahasa Indonesia yang bersifat baku serta telah melalui proses kurasi untuk memastikan kesesuaian dengan kebutuhan pengujian. *Corpus* penelitian ini diperoleh dari beberapa sumber terpercaya, dengan kriteria sebagai berikut:

- Berbahasa Indonesia sesuai kaidah ejaan yang berlaku (EYD/EBI).
- Memiliki struktur kalimat baku dan terstandar.
- Relevan untuk konteks aplikasi NLP dalam sektor pendidikan, pemerintahan, dan dokumentasi resmi.

Beberapa sumber data yang digunakan antara lain:

1. Undang-Undang Dasar Negara Republik Indonesia Tahun 1945 dan Peraturan Perundang-undangan.

   Digunakan karena teks hukum memiliki struktur bahasa baku yang ketat dan sering menjadi objek utama dalam pengolahan teks formal untuk kepentingan pencarian informasi hukum dan digitalisasi arsip (Amien, 2023).

2. Artikel Akademik atau Jurnal Ilmiah Nasional

   Artikel ilmiah memiliki konsistensi dalam penggunaan tata bahasa baku dan terminologi khusus, sehingga relevan untuk menguji kinerja stemming pada dokumen pendidikan dan penelitian. Pemrosesan teks akademik membutuhkan tingkat akurasi tinggi karena kesalahan dalam proses stemming dapat memengaruhi pemahaman makna dan relevansi dokumen dalam sistem pencarian dan klasifikasi dokumen ilmiah (Rizki et al., 2023).

3. Publikasi Resmi dari Kementerian atau Lembaga Negara

   Siaran pers, artikel edukasi, atau panduan resmi dari lembaga pemerintahan dipilih karena mencerminkan penggunaan bahasa formal dalam komunikasi publik resmi (Abidin et al., 2024).

Pemilihan korpus ini didasarkan pada kebutuhan untuk menguji algoritma *stemming* dalam konteks penggunaan yang memerlukan tingkat akurasi tinggi dan ketepatan struktur bahasa. Teks dari domain hukum, akademik, dan dokumen resmi dipilih karena:

- Memiliki struktur morfologi dan sintaksis yang konsisten dan sesuai kaidah, sehingga memudahkan proses evaluasi algoritma secara lebih objektif (Carolina et al., 2024).
- Berisiko tinggi jika mengalami kesalahan *stemming*, karena perubahan bentuk kata dalam teks resmi dapat mengubah makna secara signifikan, terutama pada dokumen hukum dan akademik (Jabbar et al., 2023).

- Belum banyak dilakukan pengujian secara spesifik menggunakan algoritma In-Idris untuk jenis teks ini, sehingga penelitian ini dapat mengisi gap yang ada.

3.3. Metode Analisis Data

Analisis data dilakukan secara kuantitatif, dengan dua fokus utama pada evaluasi algoritma yang sedang dibandingkan:

1. Evaluasi Kinerja Stemming

   Evaluasi dilakukan menggunakan tiga aspek utama:

   a. Akurasi

      Akurasi dihitung dengan cara membandingkan hasil *stemming* dari masing-masing algoritma terhadap daftar kata dasar (gold standard). Rumus akurasi sebagai berikut:

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

   b. Efisiensi waktu

      Mengukur waktu yang diperlukan algoritma untuk memproses seluruh dataset.

   c. Jenis Kesalahan

      Klasifikasi kesalahan dibagi menjadi dua:

      - *Over-stemming* (terlalu banyak memotong kata hingga akar kata hilang makna).
      - *Under-stemming* (imbuhan tidak sepenuhnya terhapus, kata tetap derivatif).

Metode evaluasi ini sesuai dengan praktik standar dalam penelitian *stemming*, yaitu mempertimbangkan akurasi, efisiensi waktu, dan error analysis sebagai bagian dari penilaian performa algoritma (Jabbar et al., 2023).

2. Pembuatan *Ground Truth*

Ground truth dalam penelitian ini dibuat untuk menentukan acuan kebenaran hasil *stemming*, yang digunakan sebagai dasar evaluasi terhadap algoritma Nazief-Adriani dan In-Idris. *Ground truth* ini menjadi komponen penting agar proses evaluasi lebih objektif dan dapat dipertanggungjawabkan.

Proses pembuatan ground truth dilakukan melalui tahapan- tahapan berikut:

a. Penggunaan Kamus KBBI sebagai Rujukan Kata Dasar

Kata dasar dalam Bahasa Indonesia ditentukan dengan mengacu pada Kamus Besar Bahasa Indonesia (KBBI) sebagai sumber utama. Setiap kata yang diolah oleh algoritma *stemming* akan dibandingkan dengan daftar kata dasar yang terdapat di dalam KBBI untuk memastikan keabsahan bentuk dasarnya.

b. Perbandingan dengan Hasil *Stemming* dari Stanza

Selain membandingkan hasil *stemming* dari Nazief-Adriani dan In-Idris dengan KBBI, dilakukan juga perbandingan dengan hasil *stemming* menggunakan Stanza, yaitu library NLP dari Stanford yang mendukung Bahasa Indonesia. Tujuan dari tahap ini adalah untuk mendapatkan referensi tambahan dalam proses verifikasi dan membantu menemukan

kasus ambigu yang mungkin sulit ditentukan secara langsung oleh anotator.

c. Validasi Manual oleh Anotator

Setelah perbandingan dilakukan, hasil *stemming* diverifikasi secara manual oleh dua anotator independen melalui sistem evaluasi berbasis web yang dikembangkan dalam penelitian ini. Jika terdapat perbedaan hasil antara algoritma dengan acuan KBBI atau Stanza, anotator menentukan secara manual apakah hasil *stemming* tersebut benar atau salah sesuai konteks kata dalam kalimat. Apabila kedua anotator berbeda pendapat, dilakukan analisa untuk mencapai konsensus sehingga hasil anotasi final bisa lebih akurat. Proses ini bertujuan untuk meminimalisasi bias dan meningkatkan kualitas data validasi.

d. Penyimpanan dan Dokumentasi

Semua hasil anotasi dan proses validasi disimpan dalam sistem berbasis web yang secara otomatis mencatat hasil, waktu validasi, anotator, dan status validasi (sepakat atau perlu diskusi). Sistem ini juga mendukung pencatatan kesalahan seperti over-stemming dan under-stemming untuk keperluan analisis lebih lanjut.

Prosedur pembuatan *ground truth* seperti ini sejalan dengan praktik standar dalam NLP yang menganjurkan kombinasi antara kamus referensi, bantuan model NLP lain sebagai pembanding, dan validasi manual agar dataset hasil anotasi memiliki kualitas tinggi (Goel et al., 2023).

3. Penggunaan Sistem Evaluasi Berbasis *Web*

Penelitian ini menggunakan sistem evaluasi berbasis *web* untuk mempermudah proses anotasi, validasi, dan dokumentasi hasil *stemming*. Sistem ini dikembangkan sebagai bagian dari upaya meningkatkan efisiensi, transparansi, dan reproducibility dalam evaluasi algoritma NLP, khususnya dalam kasus stemming Bahasa Indonesia:

a. Menyederhanakan Proses Anotasi

Anotator dapat langsung melihat hasil *stemming* dari algoritma Nazief-Adriani dan In-Idris melalui antarmuka web. Sistem memungkinkan anotator untuk mengoreksi hasil jika terdapat kesalahan dan mencatat jenis kesalahannya (misalnya *over-stemming* atau *under-stemming*).

b. Mempermudah Validasi dan Konsensus

Sistem mendukung proses validasi dua tahap dengan pencatatan hasil anotasi oleh dua orang validator. Apabila terdapat perbedaan hasil, sistem menyediakan fitur diskusi atau adjudikasi langsung melalui antarmuka web sehingga mempermudah penyelesaian perbedaan secara transparan.

c. Penyimpanan Otomatis dan Dokumentasi Kesalahan

Setiap proses anotasi dan validasi tersimpan secara otomatis ke dalam basis data. Sistem ini mencatat waktu validasi, nama anotator, status verifikasi, dan jenis kesalahan yang ditemukan. Dengan demikian, proses evaluasi dapat dilacak kembali (traceable) dan dapat diulang oleh peneliti lain (reproducible).

d. Analisis Kesalahan Secara Interaktif

Sistem dilengkapi fitur untuk mengelompokkan jenis kesalahan yang sering terjadi sehingga mempermudah analisis secara statistik. Hal ini memungkinkan peneliti untuk melihat pola error dari masing-masing algoritma secara langsung.

Penggunaan sistem interaktif seperti ini sudah menjadi praktik yang dianjurkan dalam penelitian NLP modern, karena dapat mengurangi beban kognitif anotator dan mempercepat proses evaluasi tanpa mengorbankan kualitas data. FALTE dan iSEA adalah contoh toolkit serupa yang digunakan dalam evaluasi teks panjang dan analisis kesalahan secara interaktif (Goyal et al., 2022), (Yuan et al., 2022).

3.4. Alur Penelitian

Alur penelitian disusun agar sistematis dan terstruktur, dengan tahapan sebagai berikut:

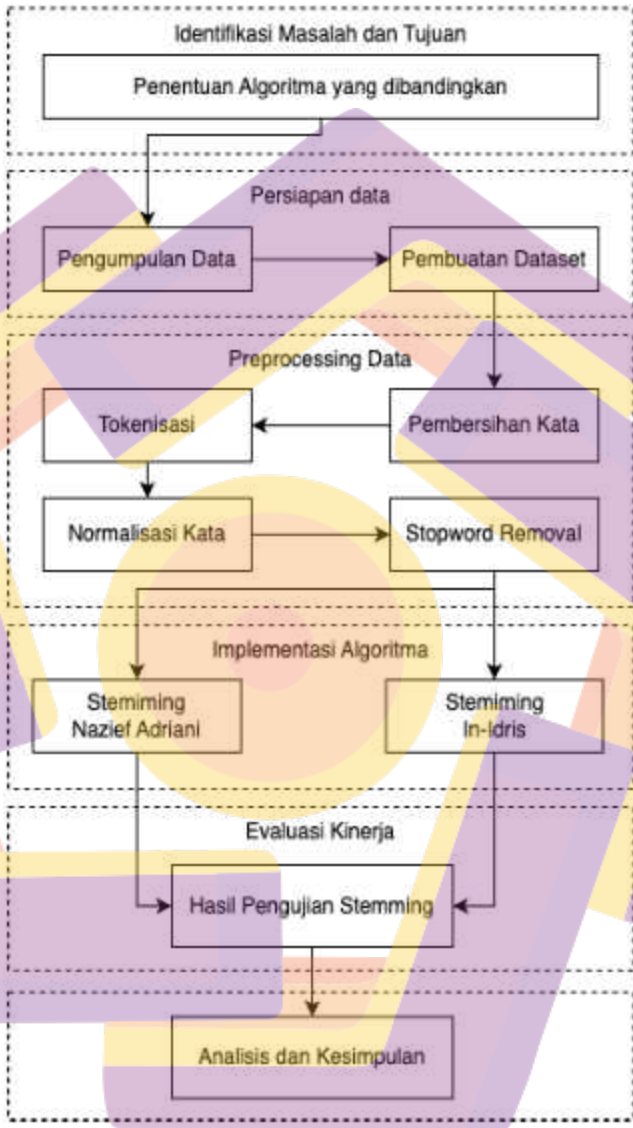
a. Identifikasi Masalah dan Tujuan
   - Menentukan algoritma yang akan dibandingkan dan metrik evaluasi.

b. Persiapan Data
   - Pengumlan data dari tiga sumber teks berbeda
   - Pembuatan Dataset

c. *Pre-Processing*
   - Pembersihan Kata, Tokenisasi, Normalisasi Kata, *Stopword Removal*

d. Implementasi Algoritma
    - Nazief-Adriani
    - In-Idris
e. Evaluasi kinerja
    - Penghitungan akurasi dan waktu proses
    - Analisis kesalahan dan performa
f. Analisis dan Kesimpulan
    - Membandingkan performa kedua algoritma secara menyeluruh
    - Menarik kesimpulan dan menyusun rekomendasi

Struktur alur ini merujuk pada pendekatan eksperimental yang umum digunakan dalam NLP, seperti dijelaskan oleh Carolina et al. (2024), yang menggarisbawahi pentingnya evaluasi metrik ganda (accuracy-efficiency) dalam pemrosesan teks lokal (Carolina et al., 2024). Gambar 3.1 memperlihatkan alur penelitian yang digunakan dalam studi ini. Struktur alur ini merujuk pada pendekatan eksperimental yang umum diterapkan dalam bidang *Natural Language Processing* (NLP), dengan penekanan pada tahapan berurutan mulai dari pemilihan data, praproses teks, penerapan algoritma *stemming*, hingga tahap evaluasi hasil. Pendekatan semacam ini memungkinkan penelitian dilakukan secara sistematis dan terukur, sehingga setiap langkah dapat ditelusuri kembali.



Gambar 3.1 - Alur Penelitian

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Identifikasi Masalah dan Tujuan

Penelitian ini berangkat dari kebutuhan untuk mengevaluasi kinerja algoritma *stemming* berbasis aturan yang selama ini menjadi tulang punggung dalam pemrosesan bahasa alami Bahasa Indonesia. Nazief-Adriani merupakan algoritma yang paling banyak digunakan, namun masih memiliki kelemahan dalam beberapa kasus seperti afiksasi berlapis dan reduplikasi. Di sisi lain, algoritma In-Idris menawarkan pendekatan yang lebih sederhana, tetapi belum banyak diuji di konteks Bahasa Indonesia, sehingga performanya masih menjadi tanda tanya.

Hingga saat ini, belum ada penelitian yang secara langsung membandingkan performa kedua algoritma ini dalam satu skema pengujian yang sama. Selain itu, evaluasi stemming di Indonesia umumnya hanya berfokus pada akurasi numerik, tanpa membahas detail jenis kesalahan yang terjadi. Padahal, dokumentasi kesalahan seperti *over-stemming* dan *under-stemming* penting untuk menemukan celah perbaikan pada aturan yang digunakan (Carolina et al., 2024).

Penelitian ini menjadi penting karena hasil stemming sangat mempengaruhi akurasi berbagai aplikasi NLP seperti pencarian informasi, klasifikasi dokumen, dan ekstraksi data. Kesalahan dalam *stemming* berisiko menyebabkan penarikan kesimpulan yang salah, terutama pada dokumen yang memuat istilah teknis dan terminologi khusus (Jabbar et al., 2023).

Oleh karena itu, penelitian ini bertujuan untuk:

- Melakukan evaluasi performa algoritma Nazief-Adriani dan In-Idris dalam pemrosesan kata berimbuhan pada korpus Bahasa Indonesia.
- Mendokumentasikan dan menganalisis jenis kesalahan yang muncul selama proses *stemming*, baik *over-stemming* maupun *under-stemming*.
- Memberikan saran perbaikan atau pengembangan aturan *stemming*, jika hasil evaluasi menunjukkan adanya kelemahan spesifik yang perlu diperbaiki agar algoritma dapat bekerja lebih optimal.

Dengan evaluasi ini, diharapkan dapat dihasilkan rekomendasi berbasis data empiris yang bermanfaat untuk pengembangan sistem pemrosesan bahasa alami berbahasa Indonesia di masa depan.

4.2. Persiapan Data

Data yang digunakan dalam penelitian ini adalah korpus berbahasa Indonesia yang bersifat baku dan memiliki struktur formal. Pemilihan data ini didasarkan pada kebutuhan untuk menguji algoritma stemming dalam konteks penggunaan yang mengharuskan ketepatan morfologi tinggi, seperti dalam dokumen hukum, akademik, dan administrasi publik. Kesalahan stemming pada teks jenis ini berpotensi menyebabkan perubahan makna yang signifikan sehingga pemrosesannya memerlukan perhatian khusus (Carolina et al., 2024). Langkah pengumpulan data antara lain:

1. Menentukan Kategori Data

   Data yang digunakan difokuskan pada teks berbahasa Indonesia yang sesuai dengan kaidah Ejaan Bahasa Indonesia (EBI), meliputi:

   - Dokumen hukum
   - Artikel ilmiah
   - Jurnal Publikasi

2. Sumber Data

   Data diperoleh dari sumber terbuka (open access) yang dapat diunduh atau diakses langsung dan bebas digunakan untuk penelitian:

   1. UUD 1945
   2. Preservasi Bahasa (Kemdikbud)
   3. Sejarah Dan Perkembangan Teknik Natural Language Processing (NLP) Bahasa Indonesia

3. Alasan Pemilihan Data

   - Struktur Bahasa Baku: Data dari dokumen resmi dan artikel ilmiah menggunakan bahasa yang sesuai kaidah, sehingga lebih relevan untuk evaluasi algoritma *rule-based*.
   - Risiko Kesalahan Semantik: Kesalahan *stemming* pada teks jenis ini dapat menyebabkan perubahan makna yang berbahaya, terutama dalam konteks hukum dan akademik.
   - Ketersediaan dan Aksesibilitas: Sumber data dapat diakses secara bebas (open access) dan legal digunakan untuk penelitian akademik (Abidin et al., 2024).

Dengan pemilihan data yang berfokus pada dokumen resmi dan artikel ilmiah berbahasa Indonesia, proses pengujian algoritma dapat dilakukan secara lebih terkontrol dan terukur. Data ini mewakili kebutuhan nyata dalam pengolahan teks formal yang memerlukan akurasi tinggi, sehingga hasil evaluasi diharapkan dapat memberikan gambaran yang lebih jelas mengenai kinerja dan kelemahan masing-masing algoritma. Langkah ini juga menjadi dasar yang penting untuk menentukan apakah diperlukan perbaikan atau pengembangan aturan lebih lanjut pada algoritma stemming yang diuji.

4.3. Pre-Processing Data

Tahap *pre-processing* merupakan bagian penting dari alur penelitian yang secara eksplisit ditunjukkan dalam Gambar 3.1. Pada diagram tersebut, pre-processing terdiri dari empat subproses utama: pembersihan kata (cleaning), tokenisasi, normalisasi kata, dan penghapusan *stopword* (stopword removal). Keempat langkah ini dirancang untuk mengubah teks mentah menjadi bentuk yang lebih terstruktur dan siap diproses oleh algoritma stemming, baik Nazief-Adriani maupun In-Idris. *Pre-processing* bukan hanya tahapan teknis, tetapi bagian integral yang menentukan kualitas hasil *stemming*. Penelitian dalam bidang NLP telah menunjukkan bahwa *pre-processing* yang baik dapat meningkatkan performa algoritma secara signifikan (Geetha et al., 2023).

1. Pembersihan Data

   Proses ini mencakup penghapusan URL, simbol, emoji, karakter non-alfabet, dan angka yang tidak relevan. Tindakan ini penting terutama pada data media sosial yang banyak mengandung elemen non-linguistik yang mengganggu pemrosesan teks. Sebagaimana dijelaskan dalam studi oleh (Fauziah et al., 2021), langkah-langkah ini merupakan dasar yang harus dilakukan dalam setiap analisis sentimen atau klasifikasi teks.

2. Tokenisasi

   Tokenisasi memisahkan teks menjadi unit-unit kata atau token. Ini merupakan tahapan kunci dalam hampir semua proses NLP karena memengaruhi akurasi ekstraksi fitur dan analisis morfologi lanjutan. Tokenisasi juga membantu memetakan struktur kalimat yang kompleks menjadi format yang dapat diproses komputer (Rizki et al., 2023).

3. Normalisasi

   Untuk menangani variasi ejaan informal, seperti "gk", "ga", atau "nggak" menjadi "tidak", dilakukan normalisasi ejaan. Langkah ini sangat krusial dalam pemrosesan data media sosial yang informal. Menurut Faturohman & Arifin (2025), normalisasi merupakan fondasi penting dalam meningkatkan kualitas data untuk analisis sentimen berbasis teks sosial (Faturohman & Arifin, 2025).

4. *Stopword Removal*

   Stopword seperti "dan", "yang", atau "atau" sering kali tidak memberikan kontribusi semantik yang signifikan dan dapat dihapus untuk efisiensi

pemrosesan. Metode penghapusan ini telah terbukti meningkatkan performa dalam klasifikasi teks seperti yang dijelaskan oleh Achsan et al. (2023) yang menggunakan pendekatan TF-IDF untuk ekstraksi *stopword* Bahasa Indonesia secara otomatis (Achsan et al., 2023).

Keempat proses ini tidak berdiri sendiri, melainkan saling terhubung dalam *pipeline pre-processing* yang utuh. Setiap tahap diproses secara berurutan dan saling memperkuat satu sama lain untuk menghasilkan korpus yang siap diolah secara objektif oleh algoritma. Dengan pre-processing yang terstandarisasi dan sejalan dengan alur dalam Gambar 3.1, maka proses implementasi algoritma di tahap berikutnya akan benar-benar mencerminkan performa aktual dari masing-masing metode stemming, tanpa bias dari kondisi awal data yang berbeda-beda.

4.4. Implementasi Algoritma

Implementasi algoritma *stemming* dalam penelitian ini dilakukan melalui pengembangan aplikasi berbasis web yang memadukan proses pemrosesan teks dengan sistem evaluasi interaktif. Platform ini dikembangkan menggunakan framework FastAPI sebagai *backend* dan Nuxt.js sebagai *frontend*. Tujuan utama sistem ini adalah untuk memfasilitasi proses pengujian dan evaluasi terhadap dua algoritma *stemming* Bahasa Indonesia, yaitu Nazief-Adriani dan In-Idris, dengan pendekatan *semi-otomatis* yang memungkinkan validasi manual oleh pengguna.

Aplikasi memiliki tiga menu utama yang dirancang untuk mendukung proses evaluasi secara lengkap dan terintegrasi, antara lain:

1. Dashboard

Menu Dashboard berfungsi sebagai tampilan utama sistem evaluasi stemming. Pada halaman ini, pengguna dapat melihat ringkasan hasil evaluasi dari kedua algoritma yang dibandingkan, yaitu In-Idris dan Nazief-Adriani. Informasi yang disajikan mencakup jumlah kata yang diproses, tingkat akurasi akhir (%), serta jumlah kesalahan *stemming* pada masing-masing dataset. Selain itu, dashboard juga menampilkan daftar file hasil stemming yang telah dianalisis dan tombol akses langsung ke visualisasi detail, sehingga pengguna dapat dengan cepat meninjau performa algoritma per korpus. Tampilan menu ini dapat dilihat pada Gambar 4.1, yang menggambarkan struktur antarmuka sistem dengan komponen warna yang membedakan hasil masing-masing algoritma.

Gambar 4.1 - Menu Dashboard

2. Menu In-Idris

Menu In-Idris merupakan salah satu antarmuka evaluasi yang disediakan dalam sistem untuk menampilkan hasil *stemming* yang dihasilkan oleh algoritma In-Idris. Pada halaman ini, pengguna dapat memasukkan teks melalui kolom input (textfield) dan menekan tombol "Proses" untuk memulai proses *stemming*. Sistem kemudian akan menampilkan hasil *stemming* dari kata-kata dalam teks tersebut dalam bentuk tabel, sebagaimana terlihat pada Gambar 4.2.

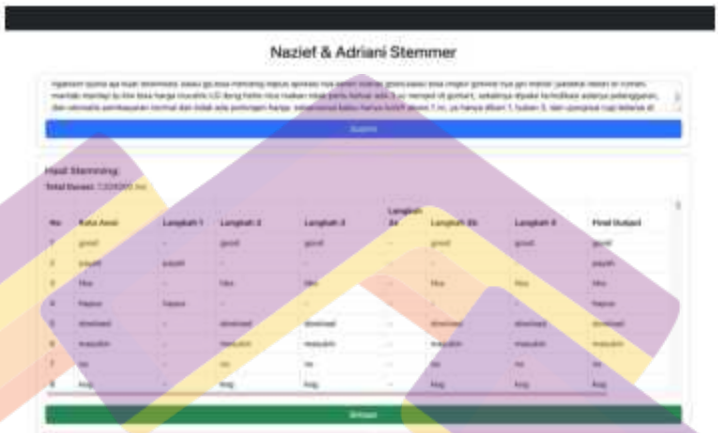
Gambar 4.2 - Menu In-Idris

Hasil tersebut disajikan dalam format per kata, dengan kolom yang memuat bentuk kata asli (kata aktual), langkah dan hasil *stemming*, serta opsi validasi manual. Pengguna dapat melakukan validasi dengan memberikan

centang pada kata-kata yang hasil *stemming*-nya dianggap benar. Setelah proses validasi selesai, pengguna menekan tombol "Simpan" untuk menyimpan data evaluasi. Data ini kemudian akan secara otomatis direkapitulasi di menu Dashboard, dan menjadi bagian dari metrik evaluasi keseluruhan. Fitur ini memungkinkan adanya penilaian manual yang melengkapi evaluasi otomatis, serta membuka ruang analisis lebih dalam terhadap ketepatan algoritma.

3. Menu Nazief & Adriani

   Menu Nazief & Adriani memiliki struktur dan mekanisme yang serupa dengan menu In-Idris. Pada halaman ini, pengguna dapat melakukan stemming terhadap teks dengan menggunakan algoritma Nazief-Adriani, yaitu algoritma berbasis kombinasi antara kamus dan aturan linguistik. Setelah pengguna memasukkan teks dan menekan tombol "Proses", sistem akan menampilkan hasil stemming dalam bentuk tabel terstruktur, sebagaimana ditampilkan pada Gambar 4.3.



Gambar 4.3 - Menu Nazief Adriani

Tabel hasil mencakup informasi seperti kata asli, langkah dan hasil stemming, serta kolom validasi manual yang memungkinkan pengguna untuk menilai akurasi hasil per kata. Proses validasi dilakukan dengan cara memberi centang pada kolom yang sesuai, dan hasil evaluasi dapat disimpan melalui tombol "Simpan". Sama seperti menu sebelumnya, data yang telah disimpan akan secara otomatis tersinkronisasi dengan menu Dashboard, sehingga dapat direkap dan dibandingkan performanya dengan algoritma lain.

Sistem ini mengadopsi pendekatan *semi-automated evaluation*, di mana proses penilaian hasil *stemming* dilakukan melalui antarmuka web yang memungkinkan pengguna untuk melakukan validasi manual terhadap keluaran algoritma. Pendekatan ini menciptakan kombinasi antara efisiensi evaluasi otomatis dan akurasi dari intervensi manusia. Validasi manual dilakukan dengan cara memberi

tanda centang pada hasil *stemming* yang dianggap benar, sehingga sistem dapat merekam koreksi langsung dari pengguna dan menyimpannya ke dalam basis data untuk keperluan analisis lebih lanjut.

Model evaluasi ini telah banyak digunakan dalam berbagai penelitian NLP yang menekankan pentingnya interaksi manusia-komputer dalam proses anotasi dan evaluasi linguistik. Validasi manual menjadi aspek krusial terutama saat menangani data sosial media yang bersifat informal, tidak baku, dan sering kali mengandung variasi bahasa lokal. Penelitian oleh (Rizki et al., 2023) menunjukkan bahwa keterlibatan pengguna dalam proses anotasi secara signifikan meningkatkan akurasi kontekstual, terutama dalam teks-teks yang berasal dari platform seperti Twitter, yang memiliki karakteristik linguistik unik dan tidak konsisten.

Evaluasi dilakukan terhadap performa dua algoritma *stemming* yang digunakan Nazief-Adriani dan In-Idris dengan menggunakan dua metrik utama, yaitu akurasi *stemming* dan waktu proses. Akurasi mencerminkan sejauh mana hasil *stemming* sesuai dengan bentuk dasar kata yang seharusnya, sedangkan waktu proses digunakan untuk mengukur efisiensi masing-masing algoritma dalam pengolahan data dalam jumlah besar. Seluruh hasil *stemming*, termasuk jumlah kata, kata yang salah di*stemming*, dan akurasi akhir, dikumpulkan secara sistematis dalam basis data melalui dashboard web. Fitur visualisasi ini dirancang agar pengguna dapat dengan mudah membandingkan performa algoritma secara menyeluruh dan membuat keputusan berbasis data terhadap efektivitas penggunaan masing-masing algoritma.

4.5. Hasil Evaluasi

Evaluasi dilakukan untuk mengetahui performa dua algoritma *stemming* berbasis aturan, yaitu In-Idris dan Nazief-Adriani, terhadap korpus berbahasa Indonesia yang telah disiapkan. Pengujian dilakukan dengan membandingkan kedua algoritma dari sisi akurasi *stemming* dan waktu pemrosesan. Selain itu, analisis terhadap jenis kesalahan juga dilakukan untuk mendokumentasikan pola error yang muncul, seperti *over-stemming*, *under-stemming*, dan *stemming* yang menghasilkan kata tidak bermakna.

Tabel 4.1 Perbandingan Hasil Evaluasi

Perbandingan Performa Algoritma In-Idris dan Nazief-Adriani

No	Nama Korpus	Jumlah Kata	Akurasi In-Idris	Durasi In-Idris (m/s)	Akurasi Nazief-Adriani	Durasi Nazief-Adriani (m/s)
1	Preservasi Bahasa	1.888	82.46%	10.844	74.35%	7.901
2	UUD 1945	2.320	84.65%	7.190	72.19%	5.514
3	Sejarah Dan Perkembangan Teknik Natural Language Processing (NLP) Bahasa Indonesia	2733	86.86%	10.249	81.18%	8.027

Tabel 4.1 dapat dilihat bahwa algoritma In-Idris menghasilkan akurasi lebih tinggi dibandingkan Nazief-Adriani pada seluruh korpus. Namun, waktu pemrosesan In-Idris cenderung lebih lama. Evaluasi lebih lanjut dilakukan dengan mencatat daftar kata yang mengalami kesalahan *stemming*, baik berupa *over-stemming*, *under-stemming*, maupun kesalahan lainnya. Beberapa kata yang sering mengalami kesalahan stemming di algoritma In-Idris antara lain:

- pemerintah (43 kali salah)
- pelestarian (30 kali salah)
- pendekatan (29 kali salah)
- pengembangan (23 kali salah)
- pemerintahan (23 kali salah)

Untuk memahami penyebab kesalahan pada algoritma In-Idris, dilakukan simulasi langkah demi langkah proses *stemming* terhadap beberapa kata.

Tabel 4.2 Simulasi kesalahan *Stemming* In-Idris
Langkah – langkah dalam proses *stemming* algoritma In-Idris

No	Kata Awal	Langkah 1	Langkah 2	Langkah 3	Langkah 4	Langkah 5	Langkah 6	Langkah 7
1	pemerintah	merintah		merintah		merintah	merintah	pemerintah
2	pelestarian		pelestarian	pelestarian		pelestarian	pelestari	pelestari
3	pendekatan	tdekatan		tdekatan		tdekatan	tdekat	pendekat
4	pengembangan	kembangan		mbangan		mbangan	mbang	pengembang
5	pemerintahan	merintahan		merintahan		merintahan	merintah	pemerintah

Dari Tabel 4.2 terlihat bahwa In-Idris melakukan pemotongan afiks secara berulang dan hierarkis, mengikuti aturan penghilangan prefiks dan sufiks. Namun, karena algoritma ini tidak menggunakan validasi kamus disetiap langkahnya, beberapa kata seperti pemerintah dan pemerintahan diproses hingga tahap yang sama dengan prefiks dihapus, lalu akhirnya dikembalikan ke bentuk awal karena tidak memenuhi

aturan lanjutannya. Hal ini mengakibatkan prosesnya cenderung bolak-balik dan bisa menyebabkan *over-stemming* jika aturan tidak berhenti di waktu yang tepat. Pada kata seperti pelestarian dan pengembangan, In-Idris cenderung mengembalikan ke bentuk dasar yang lebih dekat dengan kata dasarnya, tetapi pada beberapa kasus justru terjadi pemotongan yang tidak sesuai konteks (misalnya, menjadi pelestari yang sebenarnya tidak diinginkan dalam semua konteks). Pada Gambar 4.4, memperlihatkan sebuah diagram alur simulasi mengenai kesalahan dalam tahapan *stemming* pada kata **"Pelestarian"**, dimana ilustrasi tersebut menunjukkan bahwa algoritma *stemming* mengalami kegagalan dan mengembalikan kata ke dalam bentuk dasarnya.

Gambar 4.5 menampilkan simulasi proses *stemming* menggunakan algoritma In-Idris. Pada contoh ini, kata "kekuatan" diproses melalui alur yang ditunjukkan pada

Gambar 4.4 – Diagram Alur Kesalahan *Stemming* In-Idris

diagram, sehingga secara bertahap imbuhan-imbuhan yang melekat pada kata diidentifikasi dan dihapus sesuai aturan yang berlaku. Simulasi ini memperlihatkan bahwa alur kerja algoritma In-Idris dapat menangani kata berimbuhan dengan tepat, sehingga menghasilkan bentuk dasar yang sesuai. Dengan demikian, Gambar 4.5 menjadi ilustrasi konkret tentang bagaimana aturan-aturan dalam algoritma diterapkan untuk mencapai hasil *stemming* yang akurat.

Gambar 4.5 – Diagram Alur berhasil Stemming In-Idris

Sementara itu, algoritma Nazief-Adriani memiliki daftar kesalahan sebagai berikut:

- penelitian (65 kali salah)
- pemerintah (43 kali salah)
- perwakilan (41 kali salah)

- pelestarian (30 kali salah)
- pendekatan (29 kali salah)

Nazief-Adriani cenderung mengalami *over-stemming*, memotong imbuhan pada kata yang seharusnya tetap utuh atau memiliki makna spesifik di konteks tertentu.

Tabel 4.3 Simulasi kesalahan *Stemming* Nazief-Adriani
Langkah – langkah dalam proses *stemming* algoritma Nazief-Adriani

No	Kata Awal	Langkah 1	Langkah 2	Langkah 3	Langkah 3a	Langkah 3b	Langkah 4
1	penelitian	penelitian	peneliti		penelit	nelit	penelitian
2	pemerintah	pemerin	pemerin		pemerin	merin	pemerin
3	perwakilan	perwakilan	perwakil		perwakil	rwakil	perwakilan
4	pelestarian	pelestarian	pelestari		pelestar	lestar	pelestarian
5	pendekatan	pendekatan	pendekat		pendekat	ndekat	pendekatan

Pada Tabel 4.3 proses penghilangan afiks dilakukan dengan memanfaatkan aturan yang lebih konservatif serta validasi terhadap daftar kata dasar. Hal ini menyebabkan sebagian kata tetap dikembalikan ke bentuk aslinya jika setelah proses pemangkasan tidak ditemukan padanan di kamus. Misalnya, kata **penelitian** dipotong menjadi **peneliti**, tetapi karena bentuk ini tidak selalu cocok untuk konteksnya, sistem mengembalikannya ke **penelitian**. Kata seperti **pemerintah**, **perwakilan**, dan **pelestarian** mengalami proses yang serupa. Meskipun ada pemotongan prefiks dan sufiks, hasil akhir seringkali kembali ke bentuk semula karena validasi kamus tidak menemukan padanan yang sesuai setelah penghapusan afiks. Hal ini menyebabkan Nazief-Adriani cenderung lebih aman terhadap *over-*

stemming, tetapi berisiko mempertahankan kata yang sebenarnya bisa di-*stemming* lebih optimal. Pada Gambar 4.6, ditunjukan sebuah alur diagram simulasi pada kesalahan proses *stemming* dengan kata "**Penelitian**", dimana gambar tersebut menunjukan ketidakberhasilan pada proses *stemming* dan mengembalikan kata seperti semula sebagai kata dasar.

Gambar 4.7 menampilkan simulasi proses *stemming* menggunakan algoritma Nazief-Adriani. Pada contoh yang ditunjukkan, kata "kebudayaan" diproses melalui alur aturan yang telah ditentukan dalam algoritma. Proses dimulai dengan identifikasi prefiks ke- dan sufiks -an, yang keduanya dihapus secara

Gambar 4.7 – Diagram Alur Benar Stemming Nazief - Adriani

sistematis. Setelah kedua imbuhan tersebut dilepaskan, algoritma melakukan pengecekan pada kata yang tersisa, yaitu "budaya". Hasil pemeriksaan dalam kamus menunjukkan bahwa "budaya" adalah bentuk kata dasar yang valid. Simulasi ini membuktikan bahwa algoritma Nazief-Adriani mampu melakukan *stemming* dengan benar pada kata yang memiliki struktur afiksasi ganda, sehingga menghasilkan kata dasar "budaya". Dengan demikian, Gambar 4.7 menjadi ilustrasi konkret tentang efektivitas algoritma dalam menangani kata berimbuhan kompleks.

Berdasarkan hasil evaluasi, dapat disimpulkan bahwa kedua algoritma memiliki kelebihan dan kekurangan masing-masing:

- Ir-Idris menghasilkan akurasi lebih tinggi pada beberapa korpus karena proses pemangkasan afiks dilakukan secara agresif. Namun, hal ini juga menyebabkan *over-stemming*, terutama pada kata yang memiliki pola morfologi kompleks atau merupakan kata berulang yang bermakna khusus.
- Nazief-Adriani lebih konservatif karena menggunakan validasi kamus, sehingga lebih aman dari over-stemming. Namun, algoritma ini masih memiliki keterbatasan, terutama jika kata yang dihasilkan tidak ditemukan di kamus, sehingga prosesnya berhenti dan kata tidak dipotong secara optimal.
- Kedua algoritma sama-sama memiliki kecenderungan salah dalam menangani reduplikasi bermakna, misalnya kata seperti "undang-undang" atau "anak-anak", yang seharusnya tidak dipotong.

- Masih terdapat banyak kasus di mana algoritma tidak sensitif terhadap konteks, sehingga hasil *stemming* tidak sesuai dengan struktur atau makna kata aslinya.

Hasil evaluasi yang telah dilakukan menunjukkan bahwa kedua algoritma *stemming* yang diuji, yaitu In-Idris dan Nazief-Adriani, memiliki kelebihan dan kelemahan masing-masing. In-Idris unggul dalam akurasi, sementara Nazief-Adriani lebih efisien dari sisi waktu pemrosesan. Namun, keduanya masih menghasilkan sejumlah kesalahan, terutama terkait dengan penanganan kata berimbuhan kompleks dan reduplikasi bermakna. Temuan ini memberikan gambaran jelas tentang posisi dan tantangan algoritma *stemming* berbasis aturan saat ini, khususnya dalam pengolahan teks berbahasa Indonesia yang bersifat baku.

4.6. Rekomendasi Perbaikan Algoritma

Berdasarkan hasil evaluasi sebelumnya, ditemukan bahwa kedua algoritma *stemming* yang diuji memiliki kelemahan dalam penanganan kata dengan imbuhan kompleks dan reduplikasi bermakna. Oleh karena itu, dilakukan dua jenis eksperimen perbaikan terhadap algoritma In-Idris dan Nazief-Adriani, yaitu:

1. Penambahan *Damerau-Levenshtein Distance* (DLD) sebagai Validasi Jarak String
2. Pengembangan Aturan Baru (Hybrid Rule Modification)

Langkah ini sejalan dengan tren NLP modern yang memadukan rule-based processing dengan pendekatan berbasis similarity untuk meningkatkan fleksibilitas dan adaptabilitas algoritma (Jabbar et al., 2023).

4.6.1 Penambahan Damerau-Levenshtein Distance (DLD)

Damerau-Levenshtein Distance (DLD) merupakan metode untuk menghitung jarak edit antar dua string dengan mempertimbangkan operasi seperti penambahan, penghapusan, penggantian, dan pertukaran karakter. Penggunaan DLD dalam proses *stemming* bertujuan untuk memberikan toleransi terhadap variasi morfologi dan mengurangi kesalahan *stemming* yang terjadi karena pemangkasan afiks yang terlalu kaku .

1. Penambahan Exception List untuk Reduplikasi Bermakna

   Exception list berisi kata-kata berulang yang memiliki makna khusus dan tidak boleh di-stemming, seperti:

   - undang-undang
   - anak-anak
   - pasal-pasal
   - permusyawaratan-perwakilan

   Dengan exception list ini, algoritma mengenali kata-kata tersebut sebagai satu kesatuan yang utuh dan tidak melakukan pemangkasan.

2. Penerapan *Damerau-Levenshtein Distance (DLD)* sebelum Validasi Kamus

   Penambahan mekanisme perhitungan jarak kemiripan string menggunakan *Damerau-Levenshtein Distance* (DLD) diterapkan untuk mengurangi kesalahan *stemming* pada kata yang seharusnya tetap dipotong tetapi belum ada di kamus. DLD adalah algoritma perhitungan jarak edit yang mempertimbangkan empat jenis operasi edit:

- Penghapusan (deletion)
- Penyisipan (insertion)
- Penggantian (substitution)
- Pertukaran posisi karakter bersebelahan (transposition)

DLD dinilai lebih akurat dibandingkan Levenshtein biasa karena mendukung transposisi, yang sering terjadi dalam proses manipulasi kata di morfologi Bahasa Indonesia (Santoso et al., 2019). Dalam penelitian ini, DLD digunakan untuk membandingkan hasil *stemming* dengan daftar kata dasar di kamus Jika kata hasil *stemming* memiliki jarak DLD di bawah ambang batas tertentu (misalnya ≤ 2), maka kata dianggap cukup dekat atau mirip dengan bentuk dasar dan proses *stemming* dinyatakan berhasil, meskipun tidak identik secara persis. Hal ini dapat membantu:

- Mengurangi kasus *under-stemming*, yaitu ketika algoritma terlalu berhati-hati sehingga gagal memotong imbuhan yang sebenarnya bisa dihilangkan.
- Meningkatkan toleransi terhadap variasi bentuk kata, terutama yang tidak secara eksplisit tercantum di kamus.

Metode Damerau-Levenshtein Distance (DLD) merupakan salah satu algoritma yang banyak dimanfaatkan dalam bidang Natural Language Processing (NLP), khususnya untuk melakukan perbandingan string. Popularitas metode ini muncul karena kemampuannya dalam memberikan tingkat toleransi kesalahan yang relatif lebih baik jika dibandingkan dengan metode perbandingan string lainnya. Berbeda dengan algoritma klasik yang

hanya memperhitungkan operasi dasar seperti penyisipan (insertion), penghapusan (deletion), dan substitusi (substitution), metode DLD memperluas cakupan analisis dengan menambahkan operasi transposisi (transposition). Fitur ini memungkinkan algoritma lebih efektif ketika berhadapan dengan kesalahan pengetikan yang sering terjadi, misalnya pertukaran posisi huruf, ataupun variasi morfologis yang umum dijumpai dalam teks. koreksi ejaan, pengenalan entitas, atau pengolahan data teks yang bersifat sensitif terhadap kesalahan penulisan.

Gambar 4.8 – Alur penerapan DLD

Proses pengujian terhadap metode DLD dalam penelitian ini dilakukan berdasarkan alur kerja yang dirancang secara sistematis. Konfigurasi alur tersebut tidak hanya memperlihatkan tahapan pemrosesan

data, tetapi juga mendeskripsikan bagaimana metode ini diimplementasikan secara terstruktur. Rangkaian proses tersebut divisualisasikan dalam Gambar 4.6, yang menjadi representasi alur penerapan algoritma dalam penelitian ini. Pengujian terhadap metode ini dilakukan dengan alur dan konfigurasi yang telah disusun secara sistematis, sebagaimana diilustrasikan pada Gambar 4.6.

Tabel 4.4 Perbandingan Hasil Evaluasi

Perbandingan Performa Algoritma In-Idris dan Nazief-Adriani

No	Nama Korpus	Jumlah Kata	Akurasi In-Idris	Akurasi In-Idris (Setelah Perbaikan)	Akurasi Nazief-Adriani	Akurasi Nazief-Adriani (Setelah Perbaikan)
1	Preservasi Bahasa	1.888	82.46% (10.844 ms)	79.4% (145853.245 ms)	74.35% (7.901 **ms**)	**70.34%** (203851.510 ms)
2	UUD 1945	2.320	84.65% (7.190 **ms**)	**76.76%** (145853.245 ms)	72.19% (5.514 **ms**)	**66.23%** (179443.456 ms)
3	Sejarah Dan Perkembangan Teknik Natural Language Processing (NLP) Bahasa Indonesia	2733	86.86% (10.249)	**57.67%** (296388.094 ms)	81.18% (8.027 **ms**)	**53.57%** (499863.188 ms)

Tabel 4.5 Simulasi kesalahan *Stemming* In-Idris
Langkah – langkah dalam proses *stemming* algoritma In-Idris

No	Kata Awal	Hasil Stemming In-Idris	Proses DLD
1	pendekatan	pendekat	pelekat
2	pengembangan	pengemban	pengembangan
3	perkembangan	perkembang	perlambang
4	terjemahan	jemah	jemah
5	memahami	memaham	temahak

Tabel 4.6 Simulasi kesalahan *Stemming* Nazief-Adriani
Langkah – langkah dalam proses *stemming* algoritma Nazief-Adriani

No	Kata Awal	Hasil Stemming Nazief-Adriani	Proses DLD
1	penelitian	nelit	gepit
2	pemerintah	pemerintah	pemerintah
3	perwakilan	rwakil	rakit
4	pelestarian	lestar	ketar
5	pendekatan	ndekat	nekat

Dari hasil pengujian tersebut dapat disimpulkan bahwa:

- Integrasi DLD yang ditujukan untuk membantu mengurangi *over-stemming* malah berdampak pada penurunan akurasi secara keseluruhan. Hal ini terjadi karena DLD memperlambat proses dengan membandingkan setiap hasil stemming terhadap seluruh kamus dengan perhitungan jarak string, yang sangat mempengaruhi waktu pemrosesan. Hal ini ditujukan pada Tabel

4.4 dimana durasi proses meningkat signifikan, misalnya pada korpus "Sejarah NLP" waktu pemrosesan mencapai 296388 ms (sekitar 296 detik) dibandingkan metode sebelumnya yang hanya belasan milidetik.

- Meskipun menggunakan DLD, beberapa kata tetap salah di-*stemming* karena DLD hanya mempertimbangkan jarak karakter, bukan konteks atau struktur morfologi. Hal ini sesuai dengan temuan dalam literatur bahwa string similarity tidak selalu efektif untuk reduplikasi dan kata berstruktur kompleks dikenali meskipun terdapat sedikit perbedaan karena pemotongan afiks.
- Tabel 4.5 memperlihatkan hasil *stemming* dari algoritma In-Idris beserta hasil proses DLD-nya. Misalnya, kata "pendekatan" dipotong menjadi "pendekat", namun hasil DLD menghasilkan kata "pelekat", yang menunjukkan adanya perubahan huruf yang cukup signifikan. Pada kata "pengembangan", hasil *stemming* masih mendekati bentuk aslinya sehingga DLD menghasilkan kata yang tetap "pengembangan", menandakan jarak edit minimal. Namun, pada kata "perkembangan", hasil DLD berubah menjadi "perlambang", menandakan kesalahan *stemming* yang cukup jauh dari makna aslinya.
- Tabel 4.6 menunjukkan hasil *stemming* dengan algoritma Nazief-Adriani dan perbandingan hasil DLD. Untuk kata "penelitian", hasil *stemming* menjadi "nelit", sedangkan DLD memberikan hasil "gepit", yang jelas merupakan bentuk yang sangat berbeda dari kata dasarnya. Ini menunjukkan adanya *over-stemming* yang berlebihan. Sebaliknya, pada

kata "pemerintah", hasil *stemming* dan proses DLD tetap mengembalikan kata "pemerintah", yang berarti proses *stemming* berjalan dengan benar. Untuk kata lain seperti "perwakilan", hasil DLD menjadi "rakit", yang memperlihatkan bahwa pemotongan imbuhan tidak dilakukan dengan akurat sehingga menghasilkan kata yang jauh dari makna aslinya

Kesalahan tetap banyak terjadi pada kata-kata yang memiliki prefiks dan sufiks berlapis, serta kata tetap (fixed terms). Hal ini menegaskan perlunya exception list dan aturan yang lebih kontekstual, bukan hanya berbasis jarak string.

4.6.2 Pengembangan Aturan Baru (Hybrid Rule Modification)

Sebagai tindak lanjut dari analisis kesalahan yang ditemukan pada algoritma In-Idris dan Nazief-Adriani, penelitian ini mengembangkan aturan *stemming* baru yang lebih komprehensif dan kontekstual. Aturan ini dirancang untuk meminimalisir kesalahan *over-stemming* maupun *under-stemming*, serta menangani masalah umum pada pemrosesan Bahasa Indonesia, seperti reduplikasi, konfiks, dan perubahan fonologi. Pengembangan aturan ini mengadopsi pendekatan *hybrid* dengan menggabungkan karakteristik dari dua algoritma terdahulu serta menambahkan tahapan baru sesuai struktur morfologi Bahasa Indonesia. Metode ini sejalan dengan rekomendasi dari penelitian terkini terkait proses *stemming* berbasis aturan yang lebih adaptif (Carolina et al., 2024), (Abidin et al., 2024).

Langkah-Langkah Hybrid Rule Stemming ini adalah sebagai berikut:

1. Cek kata di Kamus

   Jika hasilnya ada di kamus, gunakan sebagai kata dasar. Jika tidak ada, maka lanjut ke langkah berikutnya.

2. Cek *Exception* List

   Jika kata terdapat dalam daftar exception (misalnya istilah tetap atau kata majemuk bermakna khusus), kata tersebut langsung dianggap sebagai kata dasar.

   - undang-undang → undang-undang
   - pemerintah → pemerintah
   - mata-mata → mata-mata

   Jika kata ditemukan di daftar *exception*, maka kata tersebut dijadikan kata dasar dan proses *stemming* selesai di langkah ini.

3. Hapus Awalan (Prefiks)

   Prefiks yang digunakan dalam aturan ini antara lain berke-, meng-, peng-, peme-, meny-, peny-, mem-, men-, pem-, pen-, ber-, ter-, per-, di-, ke-, se-, me-, dan pe-.

   Contoh:

   - berlari → ber- → lari
   - memukul → mem- → mukul
   - menari → men- → nari
   - diambil → di- → ambil
   - seorang → se- → orang

4. Cek kata di Kamus (Langkah 1)
5. Cek Aturan Fonologi

   Beberapa prefiks menyebabkan perubahan fonologi pada huruf pertama kata dasar. Aturan fonologi yang digunakan adalah:

   - Jika awalan mem-, pem-, peme- diikuti vokal, huruf awal p pada kata dasar sering hilang dan perlu dikembalikan.
   - Pada awalan men-, pen-, jika diikuti vokal, huruf t di kata dasar biasanya hilang sehingga perlu dipulihkan.
   - Untuk awalan meng-, peng- diikuti vokal, huruf k di awal kata dasar juga sering hilang.
   - Sementara itu, pada awalan meny- dan peny-, huruf s di awal kata dasar digantikan, sehingga saat *stemming* perlu dikembalikan ke bentuk aslinya.

   Contoh kata yang mengalami perubahan fonologi adalah sebagai berikut:

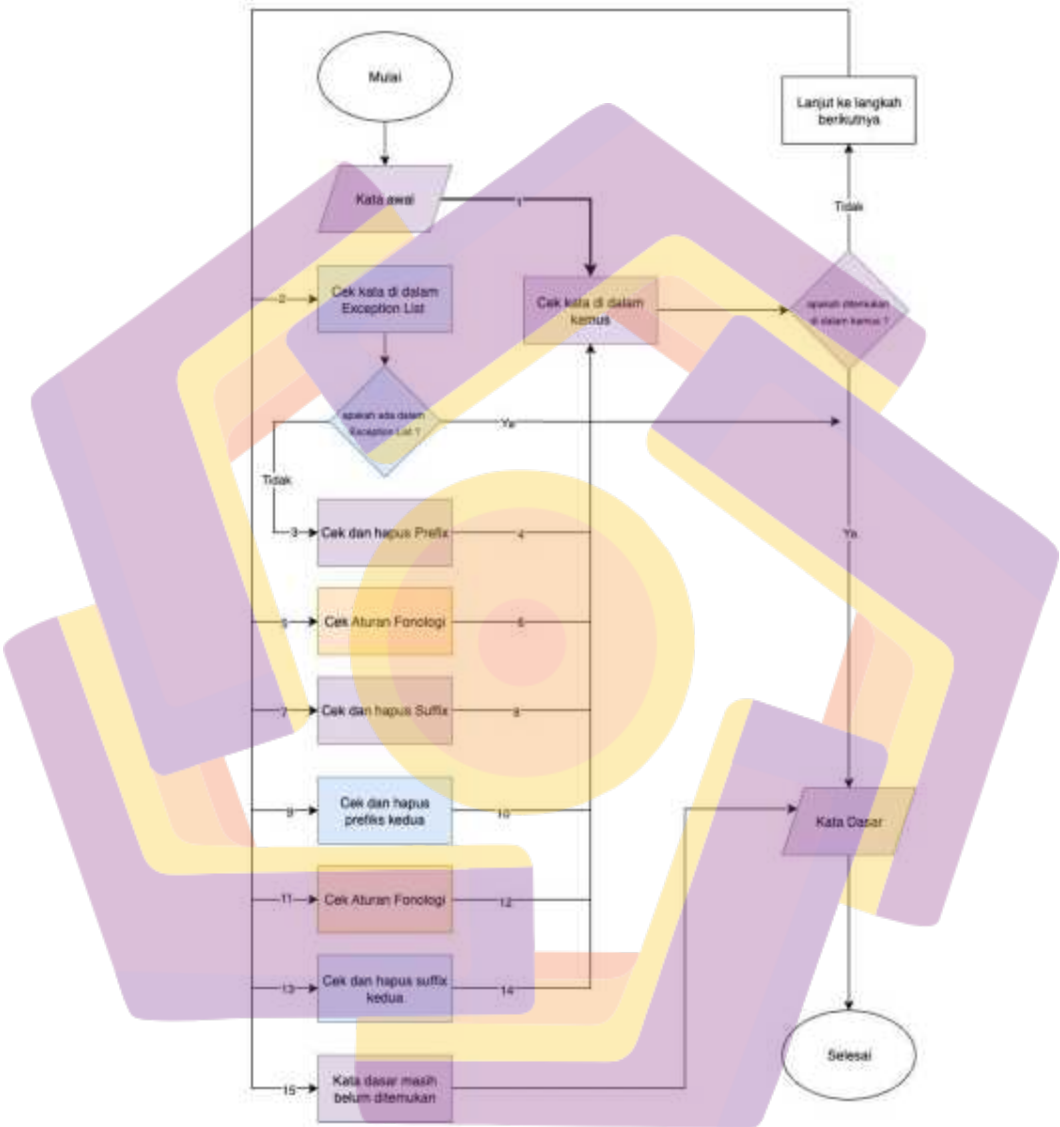
   - memukul → mukul → pukul
   - menyapu → nyapu → sapu
   - menari → nari → tari
6. Cek kata di Kamus (Langkah 1)
7. Hapus Akhiran (Sufiks)

   Terdapat beberapa sufiks atau akhiran yang umum digunakan untuk membentuk kata turunan. Sufiks tersebut antara lain: -kan, -man, -nya, -wati, -wan, -an, -is, dan -i.

   Contoh:

- berikan → -kan → beri
- seniman → -man → seni
- bukunya → -nya → buku
- tulisan → -an → tulis
- idealis → -is → ideal
- dermawan → -wan → derma

8. Cek kata di Kamus (Langkah 1)
9. Hapus prefiks kedua
    - keberpihakan → ke- ber- … -an → pihak
    - pemenggalan → pe- me- … -an → penggal
    - kepemimpinan → ke- pem- … -an → pimpin
10. Cek kata di Kamus (Langkah 1)
11. Cek Aturan Fonologi (Langkah 6)
12. Cek kata di Kamus (Langkah 1)
13. Hapus sufiks kedua
    - jabatannya → -an -nya → jabat
    - kesepakatannya→ -an -nya → sepakat
14. Cek kata di Kamus (Langkah 1)
15. Jika kata dasar masih tidak ditemukan, maka jadikan kata awal sebagai kata dasar.



Gambar 4.9 – Alur penerapan Hybrid Stemming

Gambar 4.9 menampilkan alur proses *hybrid stemming* yang diusulkan dalam penelitian ini. Pada diagram tersebut, setiap warna digunakan untuk menandai sumber aturan yang menjadi acuan dalam sistem. Bagian yang ditandai dengan warna **ungu** merepresentasikan aturan hasil gabungan antara algoritma In-Idris dan Nazief-Adriani, yang menjadi inti dari pendekatan hibrid. Sementara itu, elemen yang ditandai dengan warna **oranye** menunjukkan aturan yang secara khusus diambil dari algoritma In-Idris, sehingga tetap mempertahankan karakteristik dasar dari algoritma tersebut. Adapun bagian yang diwarnai dengan **biru** menggambarkan aturan tambahan yang diusulkan oleh penulis sendiri, sebagai bentuk kontribusi baru untuk memperbaiki kelemahan kedua algoritma terdahulu. Dengan penggabungan ini, Gambar 4.9 tidak hanya memperlihatkan integrasi dua algoritma *stemming* yang sudah ada, tetapi juga menekankan inovasi dalam bentuk aturan baru. Tujuannya adalah menghasilkan sistem *stemming* yang lebih adaptif, akurat, serta mampu menangani variasi morfologi Bahasa Indonesia dengan lebih baik.

Setelah menyusun aturan algoritma Hybrid Stemming yang menggabungkan mekanisme *exception list*, penyesuaian aturan fonologi, pemrosesan prefiks dan sufiks ganda, serta penanganan konfiks, dilakukan pengujian terhadap tiga korpus berbeda untuk mengukur kinerja algoritma:

1. Korpus Preservasi Bahasa
2. Korpus Sejarah dan Perkembangan Natural Language Processing (NLP)
3. Korpus Undang-Undang Dasar (UUD) 1945

Pengujian ini bertujuan untuk mengetahui seberapa efektif rule *hybrid* yang dikembangkan dibandingkan dua *stemmer* eksisting, yaitu In-Idris dan Nazief-Adriani.

Tabel 4.7 Perbandingan Hasil Evaluasi

Perbandingan Performa Algoritma In-Idris dan Nazief-Adriani, Hybrid

No	Nama Korpus	In-Idris	Nazief-Adriani	Hybrid Stemmer
1	Preservasi Bahasa	82.46% (10.844 ms)	74.35% (7.901 ms)	92.21% (12.750 ms)
2	Sejarah Dan Perkembangan Teknik Natural Language Processing (NLP) Bahasa Indonesia	86.86% (10.249 ms)	82.17% (8.027 ms)	93.85% (32.957 ms)
3	UUD 1945	85.75% (8.450 ms)	74.80% (8.965 ms)	90.39% (17.689 ms)

Dari Tabel 4.7, Hybrid Stemmer menunjukkan performa terbaik dibandingkan dengan In-Idris dan Nazief-Adriani pada semua korpus yang diuji. Hybrid Stemmer mampu meningkatkan akurasi *stemming* secara signifikan, terutama pada dokumen dengan struktur morfologi yang kompleks dan banyak mengandung istilah majemuk. Pada korpus Preservasi Bahasa, Hybrid Stemmer mencapai akurasi sebesar 92.21%, lebih tinggi sekitar 10% dibandingkan dengan In-Idris (82.46%) dan 18% lebih baik dibandingkan dengan Nazief-Adriani (74.35%). Pada korpus Sejarah NLP, Hybrid Stemmer meraih akurasi tertinggi yaitu 93.85%, unggul signifikan dibandingkan dua *baseline* yang masing-masing mencatatkan akurasi 86.86%. Sedangkan pada korpus UUD 1945, Hybrid Stemmer juga menunjukkan performa superior dengan akurasi 90.39%, dibandingkan dengan In-Idris yang memperoleh 85.75% dan Nazief-Adriani yang hanya mencapai 74.80%. Namun

demikian, perlu dicatat bahwa waktu pemrosesan Hybrid Stemmer relatif lebih lama dibandingkan dua metode lainnya. Hal ini disebabkan oleh kompleksitas aturan hybrid, penggunaan pengecekan kamus ganda di setiap langkah, serta proses normalisasi fonologi. Meski begitu, penambahan waktu pemrosesan ini masih dapat diterima dalam konteks aplikasi pada teks formal, di mana akurasi lebih diutamakan dibandingkan kecepatan. Tabel 4.8 merupakan hasil simulasi dari kata yang tidak berhasil di *stemming*. hal tersebut terjadi kompleksitaas imbuhan yang tidak terdeteksi oleh aturan yang telah dibuat, dan yang kedua dikarenakan angka yang dieja dengan menggunakan huruf yang tidak ada di dalam kamus.

Gambar 4.10 menampilkan simulasi proses *stemming* menggunakan algoritma Hybrid Stemmer. Pada contoh yang ditunjukkan, kata "keberagaman" diproses dengan melalui tahapan penghapusan imbuhan sesuai alur aturan yang berlaku dalam algoritma hybrid. Proses diawali dengan identifikasi prefiks ke- yang kemudian dihapus, menghasilkan kata "beragaman". Selanjutnya, algoritma kembali mendeteksi adanya prefiks ber- pada kata tersebut, sehingga dilakukan penghapusan kedua terhadap prefiks tersebut. Hasil sementara yang diperoleh adalah kata "ragaman". Pada tahap berikutnya, sistem mengenali sufiks -an yang masih melekat, lalu menghapusnya sehingga tersisa kata "ragam". Dengan demikian, Gambar 4.9 menjadi ilustrasi konkret keberhasilan pendekatan hybrid dalam menangani kata berimbuhan kompleks seperti "keberagaman", dan membuktikan keunggulannya dibandingkan algoritma yang hanya menghapus satu lapisan imbuhan.

Gambar 4.10 – Diagram Alur Benar Hybrid Stemming

Gambar 4.11 menampilkan simulasi proses *stemming* menggunakan algoritma Hybrid Stemmer pada kata "kependudukan". Hasil *stemming* yang diperoleh adalah "duduk", padahal kata dasar yang seharusnya adalah "penduduk". Kasus ini menunjukkan terjadinya *overstemming*, yaitu kondisi ketika algoritma menghapus imbuhan secara berlebihan sehingga menghasilkan kata dasar yang keliru. Meskipun algoritma hybrid yang diusulkan telah mampu menangani prefiks maupun sufiks secara berurutan dengan cukup baik, contoh pada gambar ini memperlihatkan bahwa kesalahan masih dapat terjadi. Hal ini menunjukkan adanya keterbatasan aturan dalam mengidentifikasi bentuk kata majemuk atau istilah tertentu yang memiliki struktur morfologi lebih kompleks. Dengan demikian, kasus *overstemming* pada kata "kependudukan" menjadi salah satu catatan penting dalam evaluasi, sekaligus menjadi dasar bagi rekomendasi pengembangan aturan tambahan pada penelitian selanjutnya.

Secara keseluruhan, penerapan *Hybrid Stemming* terbukti memberikan peningkatan kinerja yang signifikan dalam proses *stemming* Bahasa Indonesia, terutama untuk teks dengan struktur linguistik kompleks dan beragam. Dengan kombinasi aturan yang lebih komprehensif, Hybrid Stemmer mampu mengurangi kesalahan *over-stemming* maupun *under-stemming* yang sering ditemukan pada metode sebelumnya. Meskipun waktu pemrosesan lebih panjang, Hybrid Stemmer menawarkan solusi yang lebih akurat untuk aplikasi pemrosesan bahasa alami berbasis teks formal di bidang pemerintahan, pendidikan, dan hukum. Penelitian ini juga membuka peluang pengembangan lebih lanjut untuk optimasi waktu proses tanpa mengorbankan akurasi.

Gambar 4.11 – Diagram Alur Salah Hybrid Stemming

BAB V

PENUTUP

5.1. Kesimpulan

Penelitian ini pada awalnya dirancang untuk mengevaluasi dua algoritma stemming populer dalam Bahasa Indonesia, yaitu Nazief-Adriani dan In-Idris, serta mengidentifikasi kekuatan dan kelemahan keduanya. Dari hasil evaluasi awal, ditemukan sejumlah keterbatasan yang cukup signifikan, terutama pada penanganan kata berimbuhan kompleks, bentuk reduplikasi, dan istilah majemuk yang kerap muncul dalam teks formal. Sebagai respons terhadap temuan tersebut, penelitian kemudian dikembangkan dengan melakukan modifikasi aturan *stemming*. Pendekatan yang digunakan adalah mengadopsi sebagian aturan dari Nazief-Adriani dan In-Idris, sekaligus menambahkan aturan baru yang mencakup pengecekan *exception list*, normalisasi fonologi, serta penghapusan prefiks dan sufiks ganda. Modifikasi ini menghasilkan sebuah algoritma baru, yaitu Hybrid Stemmer, yang terbukti mampu meningkatkan akurasi secara signifikan dibandingkan kedua algoritma dasar.

Pengujian dilakukan pada tiga *corpus* formal berbahasa Indonesia, yaitu Preservasi Bahasa, Sejarah NLP, dan UUD 1945. Hasil eksperimen menunjukkan bahwa Hybrid Stemmer memberikan peningkatan akurasi pada seluruh skenario pengujian, dengan rata-rata perbaikan lebih dari 10% dibandingkan dengan In-Idris dan Nazief-Adriani. Walaupun waktu pemrosesan menjadi lebih lama akibat kompleksitas aturan tambahan, peningkatan akurasi ini membuktikan bahwa

pendekatan hybrid lebih efektif digunakan pada teks formal yang memiliki struktur baku dan konteks jelas, seperti dokumen akademik maupun peraturan resmi. Namun demikian, hasil simulasi yang ditunjukkan pada Gambar 4.11 memperlihatkan bahwa algoritma hybrid masih menghadapi tantangan pada kasus tertentu, misalnya *overstemming*. Pada contoh kata "kependudukan" yang diproses menjadi "duduk" (ekspektasi "penduduk"), terlihat bahwa meskipun algoritma mampu menangani prefiks maupun sufiks secara berurutan, kesalahan tetap dapat terjadi. Kondisi ini mengindikasikan bahwa aturan yang ada masih belum sepenuhnya mampu mengakomodasi kompleksitas morfologi kata majemuk dalam Bahasa Indonesia.

Penelitian ini juga menghadapi keterbatasan pada aspek ketersediaan dataset. Sebagian besar penelitian terdahulu mengenai *stemming* tidak menyediakan *corpus* secara terbuka, sehingga perbandingan hasil akurasi tidak dapat dilakukan secara langsung dengan basis data yang sama. Perbedaan hasil antara penelitian ini dan penelitian terdahulu sangat mungkin dipengaruhi oleh variasi *corpus* yang digunakan. Namun demikian, penelitian ini telah berupaya menjaga objektivitas dengan menyusun corpus seragam yang digunakan untuk menguji semua algoritma pembanding. Kondisi ini sekaligus menegaskan pentingnya ketersediaan dataset terbuka (*open benchmark*) dalam penelitian NLP Bahasa Indonesia agar penelitian di masa depan dapat dilakukan dengan evaluasi yang lebih konsisten.

Secara keseluruhan, penelitian ini tidak hanya memberikan kontribusi dalam bentuk evaluasi algoritma *stemming*, tetapi juga berhasil mengusulkan serta mengimplementasikan algoritma hybrid yang lebih akurat untuk Bahasa Indonesia.

Dengan demikian, penelitian ini diharapkan dapat menjadi pijakan bagi penelitian lanjutan, baik dalam pengembangan algoritma *stemming* yang lebih adaptif, penanganan kasus *overstemming* melalui integrasi *exception list* yang lebih komprehensif, maupun dalam penerapannya pada teks informal seperti media sosial dan percakapan sehari-hari.

5.2. Saran

Berdasarkan hasil penelitian dan pengembangan algoritma Hybrid Stemmer yang dilakukan, terdapat beberapa saran untuk pengembangan dan penelitian selanjutnya agar hasil yang diperoleh dapat lebih optimal dan aplikatif di berbagai konteks:

1. Pengujian pada Korpus Non-Formal

   Penelitian ini masih terbatas pada korpus berbahasa Indonesia yang bersifat formal seperti dokumen pemerintahan, artikel ilmiah, dan teks akademik. Untuk mendapatkan gambaran yang lebih komprehensif, algoritma Hybrid Stemmer perlu diuji lebih lanjut pada korpus berbahasa non-formal, seperti percakapan media sosial, chat, dan forum daring. Pengujian ini penting karena struktur morfologi pada teks non-formal biasanya lebih dinamis, sering mengandung slang, singkatan, atau penulisan tidak baku.

2. Penambahan Variasi Korpus Pengujian

   Di masa depan, pengujian sebaiknya dilakukan dengan menggunakan korpus yang lebih besar dan lebih beragam dari berbagai domain, seperti korpus kesehatan, korpus berita, korpus hukum, serta teks domain spesifik

lainnya. Hal ini untuk memastikan bahwa algoritma memiliki generalisasi yang baik terhadap berbagai konteks penggunaan.

3. Optimasi Waktu Eksekusi

   Meskipun akurasi Hybrid Stemmer meningkat secara signifikan, waktu pemrosesan masih perlu dioptimalkan agar lebih efisien. Salah satu saran adalah dengan mengimplementasikan teknik parallel processing atau menggunakan struktur data yang lebih cepat seperti Trie untuk pencarian prefiks dan sufiks.

4. Penyempurnaan *Exception List* dan Kamus Dasar

   Exception list dan kamus dasar yang digunakan saat ini masih bersifat terbatas. Penelitian lanjutan perlu menyempurnakan daftar tersebut agar mencakup lebih banyak istilah khusus, serapan asing, dan kata majemuk bermakna khusus yang sering digunakan dalam Bahasa Indonesia kontemporer.

Penelitian ini telah berhasil mengevaluasi dan memodifikasi algoritma *stemming* untuk Bahasa Indonesia dengan mengusulkan pendekatan *hybrid* berbasis rule. Hasil pengujian menunjukkan bahwa metode ini mampu meningkatkan akurasi secara signifikan pada teks formal, meskipun dengan konsekuensi peningkatan waktu pemrosesan. Temuan ini diharapkan dapat menjadi dasar pengembangan sistem NLP berbahasa Indonesia yang lebih akurat dan adaptif. Dengan memperluas cakupan pengujian dan melakukan optimasi lebih lanjut, algoritma yang dikembangkan berpotensi menjadi kontribusi nyata bagi perkembangan teknologi pemrosesan bahasa Indonesia di berbagai bidang aplikasi.

DAFTAR PUSTAKA

Abidin, Z., Junaidi, A., & Wamiliana. (2024). Text Stemming and Lemmatization of Regional Languages in Indonesia: A Systematic Literature Review. *Journal of Information Systems Engineering and Business Intelligence*, *10*(2), 217–231. https://doi.org/10.20473/jisebi.10.2.217-231

Achsan, H. T. Y., Suhartanto, H., Wibowo, W. C., Dewi, D. A., & Ismed, K. (2023). Automatic Extraction of Indonesian Stopwords. *International Journal of Advanced Computer Science and Applications*, *14*(2), 166–171. https://doi.org/10.14569/IJACSA.2023.0140221

Amien, M. (2023). *Sejarah dan Perkembangan Teknik Natural Language Processing (NLP) Bahasa Indonesia: Tinjauan tentang sejarah, perkembangan teknologi, dan aplikasi NLP dalam bahasa Indonesia*. *2007*, 1–7. http://arxiv.org/abs/2304.02746

Arif Siswandi, A., Permana, Y., & Emarilis, A. (2021). Stemming Analysis Indonesian Language News Text with Porter Algorithm. *Journal of Physics: Conference Series*, *1845*(1). https://doi.org/10.1088/1742-6596/1845/1/012019

Carolina, V. P., Utami, E., & Yaqin, A. (2024). Exploring Stemming Techniques in Ambon Malay Languages: A Systematic Literature Review. *Jambura Journal of Informatics*, *6*(1), 01–13. https://doi.org/10.37905/jji.v6i1.24954

Faturohman, M. I., & Arifin, M. (2025). *TEXTBLOB-BASED SENTIMENT ANALYSIS OF TABUNGAN PERUMAHAN RAKYAT (TAPERA) POLICY : A PUBLIC PERCEPTION STUDY*. *20*(1), 11–20.

Fauziah, Y., Yuwono, B., & Aribowo, A. S. (2021). Lexicon Based Sentiment Analysis in Indonesia Languages : A Systematic Literature Review. *RSF Conference Series: Engineering and Technology*, *1*(1), 363–367. https://doi.org/10.31098/cset.v1i1.397

Geetha, D. V, Gomathy, D. C. K., Vallab Yaratha Yagn, M. D. S. D., & Praneesh, S. (2023). the Role of Natural Language Processing. *Interantional Journal of Scientific Research in Engineering and Management*, *07*(11), 1–11. https://doi.org/10.55041/ijsrem27094

Goel, A., Gueta, A., Gilon, O., Liu, C., Erell, S., Nguyen, L. H., Hao, X., Jaber, B., Reddy, S., Kartha, R., Steiner, J., Laish, I., & Feder, A. (2023). LLMs Accelerate Annotation for Medical Information Extraction. *Proceedings of Machine Learning Research*, *225*, 82–100.

Goyal, T., Li, J. J., & Durrett, G. (2022). FALTE: A Toolkit for Fine-grained Annotation for Long Text Evaluation. *EMNLP 2022 - 2022 Conference on Empirical Methods in Natural Language Processing, Proceedings of the*

Demonstrations Session, 351–358. https://doi.org/10.18653/v1/2022.emnlp-demos.35

Jabbar, A., Iqbal, S., Tamimy, M. I., Rehman, A., Bahaj, S. A., & Saba, T. (2023). An Analytical Analysis of Text Stemming Methodologies in Information Retrieval and Natural Language Processing Systems. *IEEE Access*, *11*(December), 133681–133702. https://doi.org/10.1109/ACCESS.2023.3332710

Jumadi, J., Maylawati, D. S., Pratiwi, L. D., & Ramdhani, M. A. (2021). Comparison of Nazief-Adriani and Paice-Husk algorithm for Indonesian text stemming process. *IOP Conference Series: Materials Science and Engineering*, *1098*(3), 032044. https://doi.org/10.1088/1757-899x/1098/3/032044

Mohemad, R., Mohd Muhait, N. N., Mohamad Noor, N. M., & Akma Mamat, N. F. (2023). A Comparative Study of Stemming Techniques on the Malay Text. *International Journal of Advanced Computer Science & Applications*, *14*(12).

Mustikasari, D., Widaningrum, I., Arifin, R., & Putri, W. H. E. (2021). Comparison of Effectiveness of Stemming Algorithms in Indonesian Documents. *Proceedings of the 2nd Borobudur International Symposium on Science and Technology (BIS-STE 2020)*, *203*, 154–158. https://doi.org/10.2991/aer.k.210810.025

Nugraha, D. S. (2024). Investigating the Unproductive Morphological Forms in Indonesian Language. *Asian Journal of Education and Social Studies*, *50*(4), 280–294. https://doi.org/10.9734/ajess/2024/v50i41330

Nugraha, D. S. (2025). Exploring Integrative Complexity in the Morphology of Indonesian Language. *South Asian Research Journal of Arts, Language and Literature*, *7*(01), 8–21. https://doi.org/10.36346/sarjall.2025.v07i01.002

Patil, C. S. (2022). NLP Assisted Text Annotation. *Interantional Journal of Scientific Research in Engineering and Management*, *06*(06), 1–10. https://doi.org/10.55041/ijsrem14651

Rizki, A. S., Aristi, N. M., Ridha, N., Zulfahri, A. F., & Wibowo, D. A. (2023). Implementation of The Indonesian Language Stemming Algorithm in Twitter Data Preprocessing. Case Study: Twitter Wargabanua and Instakalsel. *Fidelity : Jurnal Teknik Elektro*, *5*(3), 175–183. https://doi.org/10.52005/fidelity.v5i3.170

Santoso, P., Yuliawati, P., Shalahuddin, R., & Wibawa, A. P. (2019). Damerau Levenshtein Distance for Indonesian Spelling Correction. *Jurnal Informatika*, *13*(2), 11. https://doi.org/10.26555/jifo.v13i2.a15698

Suci, F. W., Hayatin, N., & Munarko, Y. (2022). in-Idris: Modification of Idris Stemming Algorithm for Indonesian Text. *IIUM Engineering Journal*, *23*(1), 82–94. https://doi.org/10.31436/IIUMEJ.V23I1.1783

Tuhpatussania, S., Utami, E., & Hartanto, A. D. (2022). Comparison of Porters Stemming Algorithm and Nazief & Adriani'S Stemming Algorithm in Determining Indonesian Language Learning Modules. *Jurnal Pilar Nusa Mandiri*, *18*(2), 203–210. https://doi.org/10.33480/pilar.v18i2.3940

Yuan, J., Vig, J., & Rajani, N. (2022). iSEA: An Interactive Pipeline for Semantic Error Analysis of NLP Models. *International Conference on Intelligent User Interfaces, Proceedings IUI*, 878–888. https://doi.org/10.1145/3490099.3511146