JURNAL RISET INFORMATIKA **P-ISSN: 2656-1743 |E-ISSN: 2656-1735**
Vol. 8, No. 3. Juni 2026 DOI: https://doi.org/10.34288/jri.v8i3.521
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

STEMMINDO: A WEB-BASED INDONESIAN STEMMING ENGINE USING ENHANCED CONFIX STRIPPING

Novi Prisma Yunita[-1], Helmi Roichatul Jannah[-2]

Informatics
Faculty of Computer Science, Amikom University Yogyakarta
novi@amikom.ac.id[1]

Informatics
Jenderal Soedirman University, Purwokerto
helmi.roichatul@unsoed.ac.id[2]

Abstract

Stemming is an essential preprocessing stage in Natural Language Processing (NLP), particularly for Indonesian, which has complex affixation patterns. Most Indonesian stemming implementations are provided as programming libraries, making them less accessible for beginners, educators, and non-programmer researchers. This study presents Stemmindo, a lightweight web-based Indonesian root word search application implementing the Enhanced Confix Stripping (ECS) algorithm using the Laravel framework. Unlike conventional stemming libraries, the system provides a real-time and modular interface that enables users to explore Indonesian morphological processing without writing program code. The novelty of this research lies in the implementation of ECS within an accessible web-based educational tool. Evaluation was conducted using affixation pattern testing, rule-based testing, and real-text evaluation. Testing on 20 affixation patterns achieved 90% accuracy, while evaluation on 100 words representing 33 derived prefix rules achieved 94% accuracy. After applying failure-handling strategies through exception lists and rule-level accommodations, the accuracy increased to 98%. Real-text evaluation was conducted using 1,742 words collected from Indonesian educational web content. After preprocessing and filtering, 564 unique words were evaluated, of which 366 stemming results were successfully matched with the corpus, while the remaining cases mainly consisted of named entities, noisy input, ambiguous forms, overstemming, and understemming. These findings indicate that the proposed system performs effectively for common Indonesian morphological patterns while remaining practical for educational and experimental NLP usage. Future work includes improving reduplication handling, expanding lexical resources, and enhancing accessibility features.

Keywords: Stemming; Indonesian Language; Root Word; Enhanced Confix Stripping; Web-based Application;

Abstrak

Stemming merupakan salah satu tahapan preprocessing yang penting dalam Natural Language Processing (NLP), khususnya untuk bahasa Indonesia yang memiliki pola afiksasi yang kompleks. Sebagian besar implementasi stemming bahasa Indonesia tersedia dalam bentuk library pemrograman, sehingga kurang mudah diakses oleh pemula, pendidik, dan peneliti tanpa latar belakang pemrograman. Penelitian ini memperkenalkan Stemmindo, sebuah aplikasi pencarian kata dasar bahasa Indonesia berbasis web yang ringan dengan mengimplementasikan algoritma Enhanced Confix Stripping (ECS) menggunakan framework Laravel. Berbeda dengan library stemming konvensional, sistem ini menyediakan antarmuka modular dan real-time yang memungkinkan pengguna mempelajari proses morfologis bahasa Indonesia tanpa perlu menulis kode program. Kebaruan penelitian ini terletak pada implementasi ECS pada sebuah tool edukasi berbasis web yang mudah diakses. Evaluasi dilakukan menggunakan pengujian pola afiksasi, pengujian berbasis aturan, dan real-text evaluation. Pengujian terhadap 20 pola afiksasi menghasilkan akurasi sebesar 90%, sedangkan evaluasi terhadap 100 kata yang mewakili 33 aturan prefiks turunan menghasilkan akurasi sebesar 94%. Setelah menerapkan strategi penanganan kegagalan melalui whitelist dan penyesuaian aturan, akurasi meningkat menjadi 98%. Real-text evaluation dilakukan menggunakan 1.742 kata yang dikumpulkan dari konten edukasi web berbahasa Indonesia. Setelah melalui proses preprocessing dan filtering, sebanyak

564 kata unik dievaluasi, dengan 366 hasil stemming berhasil ditemukan dalam corpus, sementara sisanya sebagian besar terdiri atas entitas nama, input tidak baku, bentuk ambigu, overstemming, dan understemming. Temuan ini menunjukkan bahwa sistem yang diusulkan bekerja secara efektif untuk pola morfologi umum bahasa Indonesia serta tetap praktis digunakan untuk kebutuhan edukasi dan eksperimen NLP. Penelitian selanjutnya dapat difokuskan pada peningkatan penanganan reduplikasi, perluasan sumber leksikal, dan pengembangan fitur aksesibilitas.

Kata kunci: Reduksi Akar Kata; Bahasa Indonesia; Akar kata; Enhanced Confix Stripping; Aplikasi Web;

INTRODUCTION

Stemming is one of the preprocessing stages in Natural Language Processing (NLP) (Almuzaini & Azmi, 2020). The application of NLP is in various sectors, such as search engines (Sawicki et al., 2023), document classification (Zhu et al., 2025)(Rianto et al., 2021), sentiment analysis (Kumar et al., 2025)(Dang et al., 2020)(Kastrati et al., 2021), machine translation (Wang et al., 2022), and information retrieval (Le et al., 2025)(Hambarde & Proenca, 2023). Stemming works to reduce words to their basic words (Jabbar et al., 2020)(Rintyarna et al., 2022). The Porter algorithm works efficiently for stemming English language (Alyousf & Alhalabi, 2025). On the other hand, in the context of Indonesian, the challenges are greater because of the agglutinative nature of Indonesian (Marlim, 2024), where a basic word can have many varying morphologies (Saddhono et al., 2023), ranging from affixes with simple patterns to affixes with complex patterns. Indonesian is a complex language due to the extensive variations in affixation and the diverse morphological structures that form word constructions (Nugraha, 2024a).

Researchers have built several Indonesian stemming techniques since the late 1990s. Nazief and Adriani (Enni Lindrawati et al., 2023) introduced a rule-based stemmer that reduces words based on Indonesian morphological rules (Prismana et al., 2021). The Nazief and Adriani algorithm is the most popular and commonly used algorithm (Darmalaksana et al., 2020) due to its rule-based characterictic and is the foundation of several algorithms that emerged afterward, namely Confix Stripping (CS) and then Enhanced Confix Stripping (ECS) (Mustikasari et al., 2021).

The CS algorithm improved upon the original by addressing several weaknesses in the Nazief-Adriani approach, such as dictionary dependence, handling of hyphenated words, and affix rule precedence (Alfian et al., 2021). Meanwhile, the Enhanced Confix Stripping (ECS) algorithm is an extension of CS that adds recursive rules, prefix-suffix combinations, and a checking step to the dictionary every time a step is executed. This dictionary checking step is intended to minimize understemming and overstemming while improving stemming validation accuracy. Sastrawi, an Indonesian stemming library (Saifullah et al., 2024), uses the principles of Nazief and Adriani, CS, and ECS in its development (Bahtiar et al., 2023). Other Indonesian stemming algorithms include the Vega algorithm, Idris algorithm, Arifin and Setiono algorithms.

Although implementations of Indonesian stemmers based on the ECS algorithm exist in the form of programming libraries like it implemented in Sastrawi, there is a lack of tools that are directly accessible through the web and designed for ease of use. Most existing solutions require installation or programming knowledge, which can be a barrier for students and non-technical users.

The topic of Indonesian language, specially morphological structures, including prefixes and suffixes, is introduced at various levels of education, from elementary to university. If the web-based stemming application is implemented, these students may benefit from an accessible and interactive tool that helps them explore how Indonesian words are formed and transformed through affixation, without requiring programming knowledge.

This study contributes through the implementation of the ECS stemming algorithm in a web-based application. The novelty of this work lies in delivering a lightweight, modular, and publicly accessible stemming tool that supports educational use and quick experimentation in Indonesian NLP. The application is designed to process single-word inputs and returns the root word, following the morphological rules of the Indonesian language.

This research utilizes a corpus of basic Indonesian words and a list of exception words. The research method includes designing a web-based system, implementing the ECS algorithm, and evaluating the accuracy of stemming on several samples of affixed words based on morphological patterns. The scope of this research is limited to applied algorithm, searching for basic words for

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i3.521
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

one input word (not a sentence or document) without integrating advanced NLP features such as part-of-speech tagging or lemmatization, nor comparison with other algorithms.

This study is expected to support Indonesian NLP research and practical learning of Indonesian morphology. The following section describes the lexical resources, ECS implementation process, and evaluation methodology used in this research.

RESEARCH METHODS

This research focuses on the implementation of the Enhanced Confix Stripping (ECS) algorithm in a web-based Indonesian stemming application. The methodology consists of system implementation, lexical resources, ECS algorithm, and evaluation of stemming performance using rule-based and real-text testing.

System Implementation

The application was implemented using Laravel with a modular service-based architecture to support ECS stemming operations and dictionary validation.

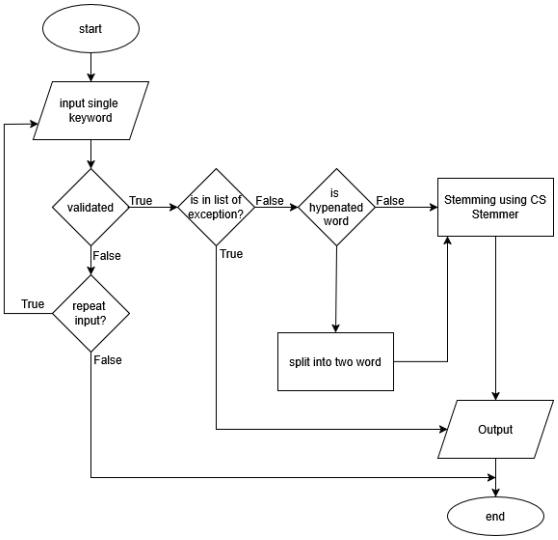

Figure 1. Flowchart of Application

The flowchart in Figure 1. outlines the stemming process, beginning from the moment a keyword is entered until the result is displayed. It includes several validation steps to determine the appropriate course of action, such as verifying the validity of the keyword, identifying whether it is an exception word, and checking for reduplication. The output is derived from two possible sources: the result of the stemming operation or the exception word list. The application provides a lightweight web-based interface consisting of an input form and result display page, allowing users to perform stemming without requiring programming knowledge.

Lexical Resources

There are two lexical resources used in this study: corpus.txt, this file serves as the reference dictionary to verify whether a word is a valid root; whitelist.txt, this file contains exception words that bypass the stemming process. The details of both files are provided in Table 1.

Table 1. List of Lexicon

File Name	Structure	Total Entries
corpus.txt	Each line contains a word followed by its category in parentheses, e.g., baca (v).	28,970 root words
whitelist.txt	Each line contains a pair in the format inflected_word=root_word, e.g., telunjuk=tunjuk.	65 exception words

The corpus containing root words obtained from https://github.com/novipeye/stemmindo/blob/main/corpus.txt, derived from Kamus Besar Bahasa Indonesia (KBBI) but filtered to contain only root words. The corpus is available in several formats, and this study uses the word-category format

The whitelist.txt contains a list of words with infixes, such as “seruling” – “suling”. ECS does not handle infixes, so the list is presented to reduce the risk of over/understemming. The whitelist was obtained from (Anistyasari & Hariadi, 2019).

Enhanced Confix Stripping Algorithm

The Enhanced Confix Stripping (ECS) algorithm is a rule-based approach that extends the original Confix Stripping method through recursive affix removal and stepwise dictionary validation. The original algorithm structure is maintained, while additional repeated dictionary checks are incorporated into the stemming process. Figure 2 presents the workflow of the ECS algorithm.

The first step of the ECS algorithm is keyword validation. If the keyword is found in the corpus, or if it consists of four characters following the original ECS stemming heuristic, it is considered a root word

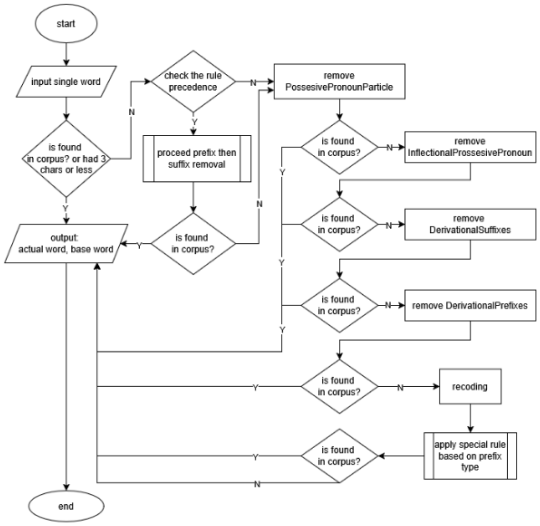

Figure 2. ECS Algorithm

ECS also applies a rule precedence checking step to handle invalid affix combinations between prefixes and suffixes. For example, the prefix "be-" cannot be combined with the suffix "-an." The list of rule precedences is presented in Table 2. Like the original algorithm, which does not define whether this check should be performed at the beginning or the end of the stemming process, this study performs rule precedence checking immediately (Pramudita et al., 2018) after keyword length validation

If the keyword satisfies the rule precedence conditions, the stemming sequence is reversed by removing prefixes before suffixes. If no root word is found, the keyword is recoded and processed again using the regular stemming procedure.

In Indonesian morphology, suffixes from the same category cannot be attached more than once. Therefore, suffix removal is performed only once. In ECS, each stemming stage is followed by a corpus validation step. If the resulting word is found in the corpus, the process stops; otherwise, the stemming process continues.

Table 2. List of Forbidden Confix Combination

Prefix	Forbidden Suffix
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an

At this stage, the algorithm removes inflectional particles (-kah, -lah, -tah, -pun), which always appear as the outermost suffixes. For example, in the word "apapun," no additional suffix can follow "-pun." Next, possessive pronouns (-ku, -mu, -nya) are removed, followed by derivational suffixes (-i, -kan, -an).

If the root word is still not found, the algorithm proceeds with the removal of derivational prefixes (ke-, se-, di-, be-, pe-, me-). When prefix removal does not produce a valid root word, recoding is performed by restoring the removed prefix before applying prefix-specific stemming rules.

If no root word is identified after all stemming rules have been applied, the word is returned to its original form.

Evaluation Method

Rule-Based Testing

To evaluate the accuracy of the stemming algorithm, 20 Indonesian affixed words were selected based on common morphological patterns. Each word was processed using the implemented system. The evaluation focused on whether the stemming result was successful or not.

An evaluation was also conducted on 33 prefix rules, with 1 to 3 words tested for each rule. If the stemming result is successful, it indicates that other words with the same pattern are also likely to succeed. Conversely, a failed stemming suggests that other words adhering to the same pattern are also prone to failure.

Real-text Evaluation

To evaluate the stemming performance in practical scenarios, Indonesian text data were collected from online news articles and educational websites. The collected text was tokenized and filtered to obtain unique affixed words. Each stemming result was manually validated against the expected root word.

The collected text underwent preprocessing consisting of case folding, symbol and numeric removal, duplicate filtering, and validation using regular expressions to retain only valid alphabetic Indonesian word forms. Although hyphens are permitted during input validation to support reduplication, the preprocessing stage used in real-text evaluation removes punctuation symbols through regex filtering, including hyphens in certain reduplicated forms.

The total number of words collected was 1,742. After the symbol and numeric removal process, the number decreased to 1,681 words. Following the

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i3.521
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

duplicate removal phase, the dataset was reduced to 709 words. Meanwhile, after the stopword removal process, the number of words became 564, which were then ready for the stemming process.

RESULTS AND ANALYSIS

This section presents the results of the system implementation and analysis of its performance. It includes an overview of the user interface and validation process, evaluation of the stemming algorithm's accuracy, and the effectiveness of failure handling strategies.

User Interface and Validation

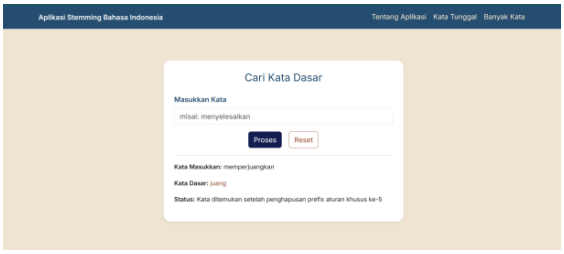

Figure 3. Homepage

Figure 3 shows the main page of the web application, where users input the word to be stemmed. The display consists of an input form.

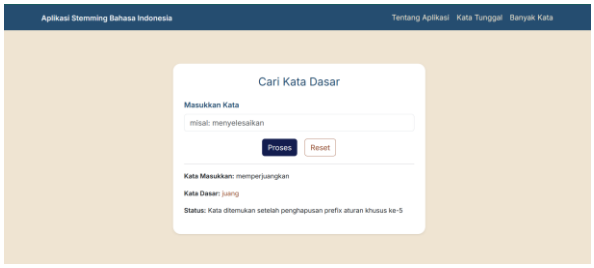

Figure 4. Result Page

Figure 4 displays the result page of the application. The input word 'memperjuangkan' is stemmed successfully to its root word 'juang', as shown in the output section.

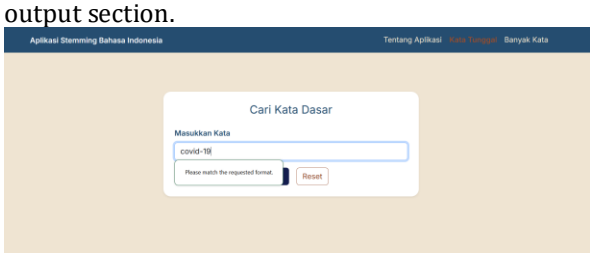

Figure 4. Client-Side Input Validation

The form input is equipped with validation. Only alphabetic characters are allowed as input, excluding numbers and symbols except hyphens used in reduplication.

Evaluation of ECS Algorithm

Rule-based testing

The evaluation was conducted by assessing the stemming results of specific affixation patterns, each evaluated once. There are 20 possible affix patterns that may occur in Indonesian words. Each affixed word was processed through the stemming algorithm. In this context, the morphological components are categorized as follows: RW (Root Word), DS (Derivational Suffix), DP (Derivational Prefix), P (Particle), and PP (Possessive Pronoun). Table 3 presents the keyword and stemming result.

In addition to using affixation patterns, the evaluation also applies to 33 derived rules. Each pattern is evaluated using one to three words, resulting in a total of 100 words being assessed. If a word is successfully stemmed, other words following the same morphological pattern are also likely to be processed correctly.

The affixation pattern evaluation achieved 18 successful cases out of 20 tested patterns (90%). Further evaluation on 100 words representing 33 derived prefix rules achieved 94 successful stemming results (94%). The 6% stemming failures occurred due to ambiguous words, and overstemming. Because of the many rules applied to affixed words, certain words end up being incorrectly stemmed when subjected to the same pattern.

Table 3. Affix Pattern and Stemming Result

Pattern	Status
RW + DS	S
RW + PP	S
RW + DS	S
DP (di) + RW + DS	S
DP (meny) + RW + DS	S
RW + P	S
DP (pem-) + RW + DS	S
DP (ber-) + RW + P	S
DP + RW + PP (-nya)	S
reduplication + DS	S
DP (di- + per-) + RW + DS (-kan)	F
DP + RW + PP	S
DP (ter-) + RW	S
DP (me-) + RW	S
RW + DS (-an) + PP (-mu)	S
DP (ber-) + RW + DS (-an)	S
DP (ke-) + RW + DS (-an)	S
DP (mem-) + RW	S
DP (di-) + RW + DS + P	S
DP (per-) + RW + DS (-an) + PP (-nya)	F

For example, in the word "berilmu", since the removal of inflectional suffixes is done at the beginning, the "mu" in berilmu is removed. This leaves the word "beril", which is an incorrect stemming result. However, it does not violate the ECS algorithm. Another example is the word "menyanyi", which produces "sanyi", an incorrect stemmed result. It should be "nyanyi". But in terms of prefix rules, this is actually correct because the "me-" prefix followed by a word beginning with "s" will be fused and replace the "s" with "ny". This rule performs effectively for the words "menyapu", "menyetujui", which yield the root words "sapu" and "setuju".

Ambiguity also arises in words like "perasa", where the stemmer outputs "asa". While "asa" is a valid root word, the intended root is "rasa". Morphologically, both interpretations are plausible: "perasa" could be derived from the prefix "pe-" and root "rasa", or from "per-" and root "asa". This highlights the inherent ambiguity present in affix combinations, which rule-based algorithms often struggle to resolve definitively.

Ambiguity is a common limitation in rule-based stemming approaches, particularly for morphologically rich languages such as Indonesian and Arabic. A related problem is overstemming. Like ambiguity, overstemming often arises from the complexity of morphological rules and the extensive variety of affix combinations found in such languages. However, in the development of this application, we aimed to accommodate various conditions in order to minimize failures. Stemming failures are addressed specifically in the following section.

Real-text Evaluation

Of the 564 processed words, 366 (64.9%) were corpus-matched, while 198 (35.1%) were corpus-unmatched. This result indicates that the ECS algorithm performs effectively for common Indonesian derivational and inflectional affixes found in real-world texts. Most successful stemming cases involve standard prefix-suffix combinations such as *ber-, ber- + -kan, di- + -kan, di- + -i, di- + -kan + -nya, ke- + -an, ke- + -an + -nya, -an, me-, mem-* which are well-covered by ECS rules.

However, stemming failures still occur in several categories, particularly named entities, ambiguous affix combinations, reduplication, and words affected by overstemming or understemming. The stemming failures can be categorized into five groups:

1. Named entities
   Words such as "Abdul", "Australia", and "Agustus" are not included in the corpus because the dictionary focuses on Indonesian root words rather than proper nouns. As a result, these words are marked as invalid although the stemming mechanism itself does not fail.
2. Reduplication
   Reduplication-related failures mainly occur because the preprocessing stage removes hyphen symbols through regular expression filtering. As a result, reduplicated forms lose their original structure before entering the stemming process, causing the resulting words to become invalid or difficult to analyze correctly by the ECS algorithm.
3. Overstemming
   Overstemming occurs when excessive affix removal produces a root that is shorter or semantically incorrect. For example, the word "diperintah" becomes "perin", indicating that recursive prefix stripping removed valid root characters.
4. Understemming
   Understemming occurs when the algorithm fails to remove all required affixes. For instance, "berkebangsaan" results in "bangsaan" instead of "bangsa", indicating incomplete suffix handling caused by rule precedence constraints.
5. Invalid or noisy words
   Invalid or noisy input occurs when the original word contains typographical errors or non-standard forms. For example, the word "berikirim" is stemmed into "ikirim" because the ECS algorithm correctly removes the prefix "ber-", but the remaining form does not correspond to a valid Indonesian root word.

The real-text evaluation produced lower validity results compared to controlled rule-based testing because real-world texts contain named entities, foreign terms, abbreviations, typographical errors, and uncommon morphological constructions that are not fully represented in the corpus.

Unlike controlled datasets, real-world corpora introduce linguistic variability that cannot always be handled using deterministic rule-based stemming alone. Examples of corpus-matched stemming results are presented in Table 4.

Table 4. Corpus-matched Stemming Result

Keyword	Stemming Result
berasimilasi	asimilasi

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i3.521
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

berdasarkan	dasar
catatan	catat
dibandingkan	banding
keberadaannya	ada
kemerdekaan	merdeka
menaklukkan	takluk
pemerintahannya	perintah

Corpus-unmatched stemming results are also identified and presented in Table 5.

Table 5. Corpus-unmatched Stemming Result

Keyword	**Stemming Result**	**Status**
abdul	abdul	person name
australia	australia	place name
berangsurangsur	angsur-angsur	expression removal, unfit
berikirim	ikirim	typographical error
berkebangsaan	bangsaan	rule precendence conflict, understemming
diperdebatkan	rdebat	understemming
diperintah	perin	overstemming
gerakan	gera	overstemming

Based on these findings, several failure-handling strategies were implemented to reduce stemming inaccuracies.

Failure Handling

Failures in stemming can occur due to several reasons: ambiguous words, overstemming, understemming, infixes, and reduplicated forms. To reduce the occurrence of such errors, two strategies are employed:

a. Ambiguous results, infix forms, and reduplicated words are addressed by registering them in the whitelist. Any word contained in this file bypasses the standard stemming procedure and directly outputs the predetermined stem as specified within the file. This mechanism is implemented to avoid conflicts between overlapping rules. Fundamentally, ambiguity is an inherent challenge in stemming for any language. Infix words are appropriately handled through whitelist due to their limited number and is unproductive (Nugraha, 2024b). Meanwhile, reduplicated words that fail to be stemmed typically represent irregular forms and are therefore sufficiently addressed through inclusion in the whitelist.
b. Rule-level accommodation (technical approach) within the stemming process. The complexity and breadth of rules in the ECS stemmer increase the likelihood of omitting certain patterns. Given that, this algorithm integrates morphological analysis, it is insufficient to rely solely on the rules outlined in the flowchart. Additional rule combinations must be formulated. For instance, in handling the removal of the suffix -i, which occurs early in the stemming sequence, there is a high risk of erroneously removing characters that are not actual suffixes. To mitigate this, compound rules are introduced. One such example: “petani” → “petan” constitutes an incorrect stemming result. Hence, a rule is added, if a word ends with -i and begins with pe-, the -i should not be removed. Similarly, for the word “menggali”, if the word ends with -i and starts with meng-, the -i should likewise be retained.

Through the application of these failure-handling strategies, the stemming evaluation has achieved a 98% success rate for the same set of keyword.

CONCLUSIONS

This study presents a web-based implementation of the Confix Stripping (ECS) algorithm using the Laravel framework for stemming Indonesian words. The application is simple, accessible through a browser, and intended to support educational and research activities in the field of Natural Language Processing (NLP).

The main contribution of this work is the development of a lightweight and modular Indonesian stemmer that can be accessed without local installation or programming knowledge. By making the ECS algorithm available as a web-based tool, this research reduces barriers to entry for NLP experimentation and provides a practical resource for learning Indonesian morphological processing.

Initial evaluations of the stemming system yielded a success rate of 90% based on affixation pattern testing and 94% across 100 words representing 33 derived prefix rules. Real-text evaluation using 1,742 collected words, which were reduced to 564 unique words after preprocessing, showed that 366 stemming results were successfully matched with the corpus. The remaining unmatched results were mainly caused by named entities, noisy input, reduplicated forms affected during preprocessing, ambiguous words, overstemming, and understemming. To reduce these failures, the system incorporates a failure-

handling mechanism using an exception word list and rule-level accommodations. After applying these strategies, the stemming accuracy on the rule-based evaluation increased to 98%, demonstrating the robustness and adaptability of the proposed system for common Indonesian morphological patterns.

Future work may include support for multiple keyword input, REST API development, improvement of infix and reduplication handling, and enhanced accessibility features.

REFERENCES

Alfian, M., Barakbah, A. R., & Winarno, I. (2021). Indonesian Online News Extraction and Clustering Using Evolving Clustering. *INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION*, *5*(3), 280–290. www.joiv.org/index.php/joiv

Almuzaini, H. A., & Azmi, A. M. (2020). Impact of Stemming and Word Embedding on Deep Learning-Based Arabic Text Categorization. *IEEE Access*, *8*, 127913–127928. https://doi.org/10.1109/ACCESS.2020.3009217

Alyousf, M., & Alhalabi, M. F. (2025). A Survey of Document Stemming Algorithms in Information Retrieval Systems. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *24*(4), 1–28. https://doi.org/10.1145/3715120

Anistyasari, Y., & Hariadi, E. (2019). ALGORITMA BARU PEMBENTUKAN KATA DASAR PADA PROSES STEMMING BAHASA INDONESIA. *Prosiding SNRT (Seminar Nasional Riset Terapan)*, 71–76.

Bahtiar, S. A. H., Dewa, C. K., & Luthfi, A. (2023). Comparison of Naïve Bayes and Logistic Regression in Sentiment Analysis on Marketplace Reviews Using Rating-Based Labeling. *Journal of Information Systems and Informatics*, *5*(3), 915–927. https://doi.org/10.51519/journalisi.v5i3.539

Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. *Electronics (Switzerland)*, *9*(3). https://doi.org/10.3390/electronics9030483

Darmalaksana, W., Slamet, C., Zulfikar, W. B., Fadillah, I. F., Maylawati, D. S. adillah, & Ali, H. (2020). Latent semantic analysis and cosine similarity for hadith search engine. *Telkomnika (Telecommunication Computing Electronics and Control)*, *18*(1), 217–227. https://doi.org/10.12928/TELKOMNIKA.V18I1.14874

Enni Lindrawati, Ema Utami, & Yaqin, A. (2023). ANoM STEMMER: Nazief & Andriani Modification for Madurese Stemming. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *7*(6), 1341–1347. https://doi.org/10.29207/resti.v7i6.5086

Hambarde, K. A., & Proenca, H. (2023). Information Retrieval: Recent Advances and beyond. *IEEE Access*, *11*, 76581–76604. https://doi.org/10.1109/ACCESS.2023.3295776

Jabbar, A., Iqbal, S., Tamimy, M. I., Hussain, S., & Akhunzada, A. (2020). Empirical evaluation and study of text stemming algorithms. *Artificial Intelligence Review*, *53*(8), 5559–5588. https://doi.org/10.1007/s10462-020-09828-3

Kastrati, Z., Dalipi, F., Imran, A. S., Pireva Nuci, K., & Wani, M. A. (2021). Sentiment Analysis of Students' Feedback with NLP and Deep Learning: A Systematic Mapping Study. *Applied Sciences*, *11*(9), 3986. https://doi.org/10.3390/app11093986

Kumar, M., Khan, L., & Chang, H. T. (2025). Evolving techniques in sentiment analysis: a comprehensive review. In *PeerJ Computer Science* (Vol. 11). PeerJ Inc. https://doi.org/10.7717/PEERJ-CS.2592

Le, D.-V.-T., Bigo, L., Herremans, D., & Keller, M. (2025). Natural Language Processing Methods for Symbolic Music Generation and Information Retrieval: A Survey. *ACM Computing Surveys*, *57*(7), 1–40. https://doi.org/10.1145/3714457

Marlim, Y. (2024). A descriptive study of affixation in Chinese and Indonesian and their morphological types. *Indonesian Journal of Applied Linguistics*, *14*(2), 273–286. https://doi.org/10.17509/ijal.v14i2.74904

Mustikasari, D., Widaningrum, I., Arifin, R., Henggal, W., & Putri, 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)*. http://tiny.cc/rootwords.

Nugraha, D. S. (2024a). Analyzing Prefix /me(N)-/ in the Indonesian Affixation: A Corpus-Based Morphology. *Theory and Practice in Language*

Studies, *14*(6), 1697–1711. https://doi.org/10.17507/tpls.1406.10

Nugraha, D. S. (2024b). 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

Pramudita, Y. D., Putro, S. S., & Makhmud, N. (2018). Klasifikasi Berita Olahraga Menggunakan Metode Naïve Bayes dengan Enhanced Confix Stripping Stemmer. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *5*(3), 269–276. https://doi.org/10.25126/jtiik.201853810

Prismana, I., Prehanto, D., Dermawan, D., Herlingga, A., & Wibawa, S. (2021). Nazief & Adriani Stemming Algorithm With Cosine Similarity Method For Integrated Telegram Chatbots With Service. *IOP Conference Series: Materials Science and Engineering*, *1125*(1), 012039. https://doi.org/10.1088/1757-899x/1125/1/012039

Rianto, Mutiara, A. B., Wibowo, E. P., & Santosa, P. I. (2021). Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00413-1

Rintyarna, B. S., Kuswanto, H., Sarno, R., Rachmaningsih, E. K., Rachman, F. H., Suharso, W., & Cahyanto, T. A. (2022). Modelling Service Quality of Internet Service Providers during COVID-19: The Customer Perspective Based on Twitter Dataset. *Informatics*, *9*(1). https://doi.org/10.3390/informatics9010011

Saddhono, K., Ermanto, Susanto, G., Istanti, W., & Sukmono, I. (2023). The Indonesian Prefix /Me-/: A Study in Productivity, Allomorphy, and Usage. *International Journal of Society, Culture and Language*, *11*(3), 115–129. https://doi.org/10.22034/ijscl.2023.1972255.2828

Saifullah, S., Dreżewski, R., Dwiyanto, F. A., Aribowo, A. S., Fauziah, Y., & Cahyana, N. H. (2024). Automated Text Annotation Using a Semi-Supervised Approach with Meta Vectorizer and Machine Learning Algorithms for Hate Speech Detection. *Applied Sciences (Switzerland)*, *14*(3). https://doi.org/10.3390/app14031078

Sawicki, J., Ganzha, M., & Paprzycki, M. (2023). The State of the Art of Natural Language Processing—A Systematic Automated Review of NLP Literature Using NLP Techniques. *Data Intelligence*, *5*(3), 707–749. https://doi.org/10.1162/dint_a_00213

Wang, H., Wu, H., He, Z., Huang, L., & Church, K. W. (2022). Progress in Machine Translation. In *Engineering* (Vol. 18, pp. 143–153). Elsevier Ltd. https://doi.org/10.1016/j.eng.2021.03.023

Zhu, H., Xia, J., Liu, R., & Deng, B. (2025). SPIRIT: Structural Entropy Guided Prefix Tuning for Hierarchical Text Classification. *Entropy*, *27*(2). https://doi.org/10.3390/e27020128