Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 3 (2025) 677 - 687 e-ISSN: 2580-0760

Minangkabau Language Stemming: A New Approach with Modified Enhanced Confix Stripping

Fadhli Almu'iini Ahda[1], Aji Prasetya Wibawa[2]*, Didik Dwi Prasetya[3], Danang Arbian Sulistyo[4], Andrew Nafalski[5]
[1,2,3,4]Elecrtrical Engineering and Informatics, Universitas Negeri Malang, Malang, Indonesia
[1,4]Informatics Engineering, Institut Teknologi dan Bisnis Asia, Malang, Indonesia
[5]Electrical Engineering, University of South Australia, Australia

[1]fadhli.almuiini.2205349@students.um.ac.id , [2]aji.prasetya.ft@um.ac.id, [3]didikdwi@um.ac.id, [4]danangarbian@gmail.com, [5]andrew.nafalski@unisa.edu.au

Abstract

Stemming is an essential procedure in natural language processing (NLP), which involves reducing words to their root forms by eliminating affixes, including prefixes, infixes, and suffixes. The employed method assesses the efficacy of stemming, which differs according to language. Complex affixation patterns in Indonesian and regional languages such as Minangkabau pose considerable difficulties for traditional algorithms. This research adopts the enhanced fixed-stripping method to tackle these issues by integrating linguistic characteristics unique to Minangkabau. This study has three phases: data acquisition, pseudocode development, and algorithm execution. Testing revealed an average accuracy of 77.8%, indicating the algorithm's proficiency in managing Minangkabau's intricate morphology. Nevertheless, constraints persist, particularly with irregular affixation patterns. Possible improvements could include adding more datasets, improving the rules for handling affixes, and using machine learning to make the system more flexible and accurate. This study emphasizes the significance of customized solutions for regional languages and provides insights into the advancement of NLP in various linguistic environments. The findings underscore the progress made in processing Minangkabau text while also emphasizing the need for further research to address current issues.

Keywords: enhanced confix stripping; minangkabau language; morphological; natural language processing; stemming

How to Cite: F. A. Ahda, Aji Prasetya Wibawa, Didik Dwi Prasetya, Danang Arbian Sulistyo, and Andrew Nafalski, "Minangkabau Language Stemming: A New Approach with Modified Enhanced Confix Stripping", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 9, no. 3, pp. 677 - 687, Jun. 2025.
Permalink/DOI: https://doi.org/10.29207/resti.v9i3.6511

Received: March 30, 2025
Accepted: June 11, 2025
Available Online: June 23, 2025

1. Introduction

Stemming, a technique in natural language processing, is employed to reduce words to their basic or root form [1]. It is a linguistic process that isolates the root word from a compound word in a phrase by separating the base word from composite words, which may consist of prefixes, infixes, and suffixes. Word stemming algorithms can be classified into various groups, each characterized by its technique and principles. Porter stemming is a technique commonly used to reduce words to their base or root form [2], [3]. Other notable stemming techniques include Snowball stemming [4], Lancaster stemming [5], Lovins stemming [6], and Indonesian stemming [7], [8]. The process of stemming has also been studied extensively in Russian and Arabic [9], [10]. while Portuguese and multilingual stemming technologies have been developed [11], [12]. The choice of stemming method depends on the specific language and the requirements of the NLP task [13]. Selecting an optimal stemming algorithm significantly influences the accuracy and effectiveness of text processing systems, as each approach has its advantages and limitations [14].

Porter stemming is simple and efficient, making it suitable for English text processing applications [15]. However, its aggressive approach can cause both excessive and insufficient stemming, reducing its suitability for tasks requiring high accuracy and completeness. Alternative stemming algorithms or more advanced NLP methods may be more appropriate depending on the language and purpose [16].

Snowball stemming has enhanced the versatility of Porter stemming by improving precision and supporting more languages [17]. Nevertheless, due to its reliance on predefined rules, it may struggle with linguistic complexities requiring very precise stemming, which points to the need for more sophisticated NLP techniques [18]. Indonesian stemming is valuable for preparing and analyzing Indonesian text but is limited when handling irregular forms and subtle contextual nuances. Greater precision and contextual awareness necessitate more advanced algorithms [19].

The Minangkabau language plays a crucial role in preserving the cultural history and identity of its people and is one of Indonesia's rich regional languages. However, integrating and managing Minangkabau in digital information technology presents significant challenges [20]. To improve the efficiency and accuracy of processing Minangkabau texts, sophisticated and context-aware stemming algorithms are essential.

Each language exhibits unique morphological variations compared to others [21]. For instance, Javanese and Minangkabau have distinct morphologies [22]. The performance of stemming algorithms depends on factors such as avoiding over-stemming, under-stemming, unchanged forms, and spelling errors. Text processing is vital for developing applications such as search engines, sentiment analysis, and root word recognition [23],[24]. Such work requires a deep understanding of word arrangement and meaning within a language [25]. The unique morphological traits of Minangkabau amplify these challenges [26]. Developing an effective stemming algorithm for Minangkabau is thus a complex task.

This research introduces a novel approach titled "Minangkabau Language Stemming: A New Approach with Modified Enhanced Confix Stripping." The goal is to overcome difficulties in processing Minangkabau by creating an improved stemming algorithm [27]. This method builds upon existing Confix Stripping technology, enhancing it with advanced techniques [28], [29]. Various studies have developed NLP algorithms for regional Indonesian languages, and this research aims to contribute significantly to preserving and advancing Minangkabau within modern information technology frameworks.

Implementing efficient stemming techniques for Minangkabau will benefit NLP applications such as Minangkabau search engines, sentiment analysis, chatbots, and more [30]. Subsequent chapters provide detailed context, objectives, methodology, and results of this study. Ultimately, this work seeks to advance NLP technologies for Minangkabau and support the preservation of its cultural identity [31].

2. Methods

2.1 Characteristics of Minangkabau Language

The Minangkabau language exhibits a rich and intricate morphological structure comprising 31 unique affixes. Six primary categories classify these affixes, each essential for word formation and modification. The language includes 11 prefixes that affix to the beginning of root words to modify their meanings or grammatical functions. Moreover, there are five combination prefixes, which consist of layered or sequential prefixes, enhancing the complexity of word formation. The language affixes four suffixes to the ends of words, which influence grammatical relationships or semantic alterations. Additionally, the language includes five infixes, which are unique affixes put within root words, contributing to a distinctive internal morphological variety. Among the less common structures are three discontinuous affixes, which consist of elements that manifest at distinct points inside the word, resulting in complex linguistic patterns. Ultimately, there exist three mixed prefixes and suffixes that amalgamate parts at both the beginning and conclusion of root words for complex word construction. This intricate affixation system emphasizes the linguistic complexity of Minangkabau and accentuates its cultural diversity, as well as the necessity for exact computer models to efficiently process and maintain the language [32].

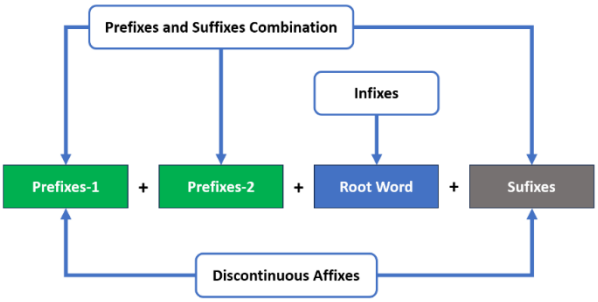

Figure 1. Flowchart morphological Minangkabau

Figure 1 is a flowchart that illustrates the morphological stages involved in word formation in the Minangkabau language. This visual representation is essential for understanding how prefixes, suffixes, and other affixes interact to generate new word forms. The process begins with the combination of two prefixes referred to as Prefix-1 and Prefix-2 which establish the initial structure of a word. At the core of this structure lies the root word, which serves as the fundamental element from which meaning is derived. Additional affixes can be attached to the root word to create new forms or alter its meaning, allowing base words to evolve into more complex expressions. One particularly notable feature in the diagram is the inclusion of discontinuous affixes, which may appear in various parts of a word, highlighting the intricate morphological patterns found in Minangkabau. Overall, the image offers valuable insight into the layered and dynamic nature of word formation in the language.

2.2 Enhanced Confix Stripping (ECS)

To determine the base or root form of a word, the improved stemming algorithm with affix removal approach iteratively removes commonly occurring prefixes and suffixes from the term. This approach operates iteratively, ensuring that any prefixes and suffixes recognized as common are systematically deleted until just the base form of the term remains [33]. This technique provides for more efficient management of morphological diversity in language, especially in languages that include multiple bound forms such as Indonesian. As demonstrated in Figure 2, this method exhibits a superior capacity in extracting the proper word root, which is vital in many linguistic applications such as text analysis and natural language processing. This affix elimination technique ensures improved accuracy in language interpretation and processing, which in turn increases the quality of data analysis outputs [34].

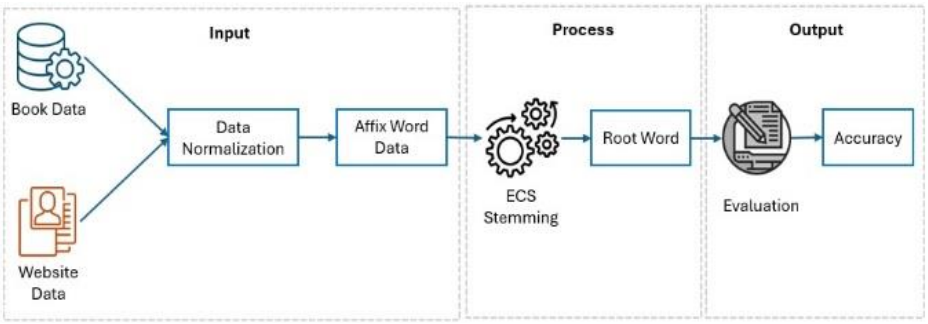

Figure 2. Architecture ECS

This approach breaks down the document stemming process into individual words, which are subsequently subjected to more complex confix stripping stemmer (ECS) methods. Figure 2 displays the overall structure used in this investigation. At the input step, the system receives a document with a .txt extension. The document contains content featuring Minangkabau rhymes and poems. In the process stage, the modified confix stripping stemmer algorithm is utilized to complete the stemming process [35]. Each word will be thoroughly verified to determine its presence in the dictionary. If the word exists in the base word dictionary, then the stemming process for the word will be skipped. Furthermore, if the word is not discovered in the base word dictionary, the upgraded confix stripping stemmer algorithm will be utilized to execute the stemming process. Once all the words in the manuscript have been analyzed, the findings of the stemming method will be given in a .txt file format that may be saved on the user's computer. All words in the original manuscript will be presented, and the stemmed words will be separated into prefix, main word, and suffix by utilizing spaces [36].

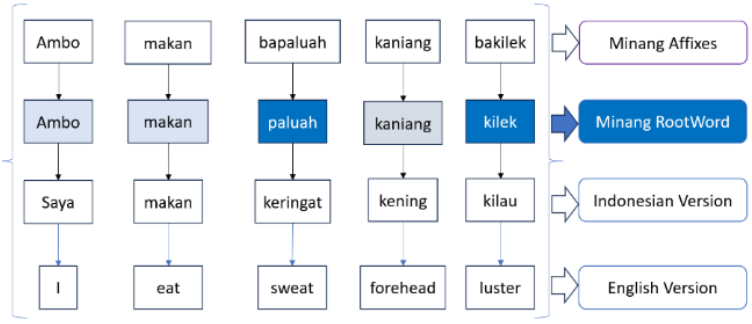

Figure 3. Illustration of the ECS Stemming Method

Figure 3 depicts the implementation of the Enhanced Confix Stripping (ECS) stemming technique on the Minangkabau lexicon, showcasing the methodical procedure of discerning and truncating words to their base forms while eliminating affixes. We examine the input, which includes Minangkabau words like "*Ambo makan bapaluah kaniang bakilek,*" to identify their morphological elements, such as prefixes, infixes, or suffixes. The ECS algorithm employs established linguistic principles to remove affixes such as *"ba-"* and *"-k,"* while maintaining the semantic integrity of the words. For example, the algorithm truncates *"bapaluah"* to *"paluah"* by removing the prefix *"ba-,"* and truncates *"bakilek"* to *"kilek"* by deleting the same prefix.

The removal of the affixes yields the root words, such as "*Ambo*" (I), "*paluah*" (sweat), "*kaniang*" (forehead), and "*kilek*" (luster). We then translate the root forms into their respective meanings in Indonesian and English, showcasing the multilingual versatility of the ECS technique. For example, "*Ambo*" translates to "*Saya*" in Indonesian and "I" in English, whereas "*kilek*" translates to "*kilau*" in Indonesian and "*luster*" in English. This procedure underscores the ECS method's accuracy in addressing Minangkabau's distinct

morphology and its applicability in natural language processing (NLP) for regional languages [37],[38].

2.3 Data Collection

The data collection method was executed meticulously by manual efforts from several sources, including Minangkabau pantun literature and other reputable web platforms. This thorough endeavor guarantees the veracity and cultural significance of the collected information. The aggregated data, including Minangkabau language terminology, was methodically arranged and rendered publicly accessible via Mendeley.data, accessible at this link https://data.mendeley.com/datasets/kch8f4smtw/1. This transparency facilitates additional research and collaboration within the linguistic and computational fields [39].

To enhance the stemming process, the gathered data was meticulously screened to identify words with affixes, essential for breaking down sentences into their basic root forms. This procedure yielded 687 distinct terms, comprising 499 words with prefixes and 188 words with suffixes, assuring no repetition for enhanced accuracy and efficiency. As part of the ECS stemming implementation, we also created a dictionary of fundamental Minangkabau phrases, containing 7,509 entries. This comprehensive vocabulary is an essential resource for comprehending and evaluating the intricate morphology of the language, facilitating the development of effective natural language processing systems specific to Minangkabau [40].

2.4 Pseudocode Enhanced Confix Stripping

To clearly illustrate the operational steps of the Enhanced Confix Stripping (ECS) algorithm, the following pseudocode outlines the core procedures involved in the stemming process. This algorithm functions by examining and removing predefined prefixes and suffixes from a word. The process is performed iteratively, prioritizing the removal of a single affix at a time either from the beginning (prefix) or end (suffix) of the word. The table below presents a structured representation of the ECS logic in pseudocode format.

Table 1. Pseudocode ECS

function Enhanced Confix Stripping Stemming (word)
Define a list of prefixes and suffixes to be stripped. prefixes = ["ma", "pa", "di", "ta", "ka", "sa"] suffixes = ["kan", "i", "an", "nyo"]
Check if the word is longer than 3 characters if length(word) > 3
Iterate through prefixes and remove them if found for prefix in prefixes. Exit the loop if a prefix is removed.
Iterate through suffixes and remove them if found for suffix in suffixes. Exit the loop if a suffix is removed.
Return word

The provided pseudocode demonstrates the Improved Confix Stripping Stemming technique, which simplifies words by removing specific prefixes and suffixes to derive their base forms [41].The algorithm begins by checking whether the input word exceeds three characters in length, ensuring that only meaningful word forms are processed. It then systematically scans a predefined list of prefixes and removes any found at the beginning of the word, terminating the loop upon successful removal. Next, the algorithm examines a list of suffixes and eliminates any that appear at the end of the word, again terminating the loop once a suffix has been removed. The final output is the word that has undergone prefix and suffix stripping, representing its simplified, stemmed form. [42]. This technique is particularly valuable in the fields of natural language processing and text analysis, as it facilitates the normalization of word variations into a consistent root form, thereby improving the efficiency and accuracy of search and analytical processes.

2.5. Evaluation

This evaluation uses a confusion matrix to measure classification accuracy and Cohen's Kappa to assess agreement between system output and human annotation.

A Confusion Matrix is a table used to evaluate the performance of a classification model by comparing predicted labels against actual labels [43].

Table 2. Confusion Matrix Table

Confusion Matrix	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

In classification tasks, prediction outcomes are grouped into four categories, as shown in Table 2. A true positive (TP) indicates a correctly identified positive case, while a true negative (TN) refers to a correctly identified negative case. A false positive (FP) occurs when a negative case is incorrectly labeled as positive, and a false negative (FN) arises when a positive case is mistakenly classified as negative. These categories are essential for evaluating the performance of stemming algorithms, particularly in detecting over-stemming and under-stemming errors in low-resource language contexts. Equations 1, 2 and 3 are the formula explained in Table 2:

$$\text{Accuracy: } \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Precision: } \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall (Sensitivity): } \frac{TP}{TP+FN} \quad (3)$$

F1 Score: Harmonic mean of Precision and Recall.

Cohen's Kappa is a statistical measure that evaluates the level of agreement between two raters, taking into account the agreement occurring by chance as shown in Equation 4 [44].

$$\text{Cohen's Kappa: } K = \frac{Po-Pe}{1-Pe} \quad (4)$$

Po refers to the observed agreement, which represents the actual proportion of agreement between raters. Meanwhile, Pe denotes the expected agreement, or the level of agreement that would be expected by chance based on the distribution of categories.

3. Results and Discussions

The Enhanced Confix Stripping (ECS) stemming algorithm utilizes a dataset of 687 affixed words, consisting of 499 words with prefixes and 188 words with suffixes. We rigorously examine these affixed terms using an extensive Minangkabau fundamental word lexicon with 7,509 entries. This comprehensive vocabulary functions as the essential resource for the algorithm to recognize and analyze the complex morphological structures characteristic of the Minangkabau language.

The ECS algorithm, through this methodical approach, has an improved capacity to fully comprehend the affixation norms and patterns specific to Minangkabau. The algorithm attains enhanced precision in recognizing accurate word stems by proficiently differentiating and extracting root words from their attached forms. This ability highlights its strength and flexibility in managing the intricate linguistic structure of regional languages. So, the ECS stemming approach greatly improves the field of natural language processing by providing a unique tool for precisely analyzing and processing texts written in the Minangkabau language. This ensures higher accuracy and linguistic significance.

3.1 Prefixes Testing

We thoroughly analyze the starting portions of Minangkabau words to determine their root forms. We examine the term "*mangintai*" and reduce it to its root word, "*intai*." We conduct this study using an extensive array of linguistic principles specifically designed for the Minangkabau language. A dataset of 499 base words with prefixes functions as the standard for administering prefix assessments.

This phase is essential for assuring the precision of the stemming process, as it allows the Enhanced Confix Stripping (ECS) method to accurately detect and separate prefixes. By doing so, the approach guarantees that the extracted root word maintains its semantic integrity, which is essential for subsequent natural language processing tasks. This prefix analysis demonstrates the algorithm's ability to manage the intricate morphological structures of the Minangkabau language with precision and reliability.

Table 3 illustrates the process of stemming words with specific markers. In computational linguistics and natural language processing, this process is important because it breaks words down to their most basic form, which makes it easier to read and understand writing. The table shows how words containing various prefixes evolved into stems during the stemming process. The base form of a word is found by taking off its affixes. This is called stemming. The "Prefix" column displays the addition of endings to words in unknown languages, such as *"sa," "man,"* and *"ba."* The "Stemming Result" box displays the base form of the stemmed word. For example, *"ikua"* comes from *"sa-ikua,"* which in English means *"tail."* Finally, the "English" column shows the stemmed results in English, which helps people who don't speak the same language understand each other.

Table 3. Prefix Stemming Result

Prefix	Affix Word	Stemming Result	English
'sa'	sa-ikua	ikua	tail
'man'	man-caliak	caliak	see
'ba'	ba-tapuak	tapuak	clap
'ka'	ka-mudiak	mudiak	homecoming
'ta'	ta-cinto	cinto	love
'many'	many-asa	sasa	regret
'mang'	mang-ecek	kecek	talk
'ta'	ta-randah	randah	low
'mam'	mam-bunuah	bunuah	kill

3.2 Suffixes Testing

Suffix analysis, the next stage of the stemming process, meticulously scrutinizes the endings of Minangkabau words to determine their root forms. The term *"sabalah"* is subjected to morphological analysis to get its base word, *"saba."* This procedure guarantees the precise identification and removal of the suffix *"lah"* without modifying the semantic essence of the word. A curated dataset of 188 base words with suffixes underpins this approach for testing purposes. The suffix-checking method is important for making the Enhanced Confix Stripping (ECS) algorithm more accurate, especially when dealing with the unique morphological features of the Minangkabau language. This phase guarantees the accurate extraction of root words while demonstrating the algorithm's resilience in handling changes in suffix application. The ECS algorithm gets better at interpreting regional language data by systematically dealing with these linguistic quirks. This makes it an important part of natural language processing for Minangkabau texts.

Table 4. Suffix Stemming Result

Suffix	Affix Word	Stemming Result	English
'lah'	bali-lah	bali	buy
'kan'	lahia-kan	lahia	born
'an'	nanti-an	nanti	later
'i'	turuik-i	turuik	follow
'nyo'	dahan-nyo	dahan	branches

Table 4 illustrates the stemming process of Minangkabau words, focusing on the identification of their root forms—a critical step in text preprocessing for linguistic analysis and natural language processing tasks. One example involves the removal of the suffix "*-lah*", as in "*bali-lah"*, which is reduced to "*bali"*, meaning "purchase." This suffix frequently functions as a marker of command or request. Another case is the suffix "*-kan*", as in "*lahia-kan*", derived from "*lahia"* ("birth"), where *-kan* typically indicates a causative or

passive verb form. The "*-an*" suffix appears in words like "*nanti-an"*, derived from "*nanti*" ("later"), and is commonly used to form nouns or adjectives. Similarly, the "*-i*" suffix, as seen in "*turuik-i*" from "*turuik"*, meaning "follow", generally denotes a transitive verb. Lastly, the "*-nyo*" suffix is illustrated in a rarer example derived from "*dahan*" ("branches"), which may represent a dialectal or informal variant. These examples highlight the morphological richness of the Minangkabau language and the complexity involved in accurately reducing inflected forms to their base roots.

In numerous NLP (Natural Language Processing) contexts, the stemming method, which reverts these words to their base form, is highly beneficial. This table presents specific instances of how prefixes and suffixes alter the meanings of words in Minangkabau. The stemming process is very important for improving the accuracy and speed of text analysis. It helps computers understand and process natural language data better by finding consistent root words even when the words' shapes are different. This can enhance the quality of data analysis outcomes, as well as promote automatic translation and more effective information retrieval.

3.3 Over-Stemming

The following table shows examples of over-stemming, where the algorithm removes too many affixes, resulting in incorrect root forms. These cases highlight the algorithm's limitations in handling complex morphological structures.

Table 5. Over-Stemming Result

Affix Word	Over-Stemming	Rootword	English
ka-mari	mar	mari	let's
di-balah	ba	balah	split
di-makan	ma	makan	eat
ka-pakan	pa	pakan	market
jan-lah	j	jan	don't
bari-lah	bar	bari	give
sado-nyo	do	sado	all
arok-an	aro	arok	hope
isi-nyo	is	isi	content
ma-nahan	ah	tahan	hold
mananti	anti	nanti	wait
paho-nyo	ho	paho	thighs
kaki-nyo	ki	kaki	foot

Based on the explanation in Table 5, Over-stemming transpires when the stemming process eliminates excessive components of a word, resulting in a root that fails to correspond with the original meaning. The terms generated by the stemming procedure in this instance do not correspond to the root of the word. An example is the Indonesian term *'jan,'* meaning *'do not.'* The term *'janlah'* originates from the Indonesian term *'jan,'* which signifies 'do not.' It omits the initial suffix *'lah'* and the subsequent suffix '*an*' twice. In Minangkabau, only the letter 'j' remains, which holds no significance. Another example is the term *'manahan,'* which originates from the root *'tahan.'* Two rules converge at this point, eliminating the prefix *'ma'* and revealing the prefix *'man'* and the suffix *'an'*, which culminate in the term *'ah.'* This term does not exist as a root word in Indonesian; it only exists as a meaningless word. The over stemming process may result in misinterpretation during linguistic analysis and natural language processing. We must accurately calibrate the stemming algorithm to maintain the integrity of the original word meaning. These examples underscore the necessity of meticulous regulation in affix removal to prevent excessive stemming that compromises the original meaning and context of the analyzed words.

Figure 4 distinctly illustrates the disparity between a successful and an unsuccessful stemming procedure, offering critical insights into the efficacy and shortcomings of the employed stemming algorithm. The red coloration on terms like '*j*' and '*aro*' signifies over-stemming, a phenomenon where the algorithm excessively eliminates components from a word, leading to an erroneous base form. For instance, the term '*janlah*,' which has the root '*jan*' and the suffix '*lah*,' erroneously reduces to '*j*' during the stemming process. Likewise, the term '*arokan*,' which ought to provide the root word '*arok*' upon the removal of the suffix 'an,' instead transforms into '*aro*,' signifying an inadequacy in the management of morphological patterns. Conversely, terms highlighted in blue, such as '*hatinyo*,' signify the efficacy of the stemming process with the Enhanced Confix Stripping (ECS) technique. The system accurately identifies the affixation pattern and generates the basic word '*hati*' as anticipated. This achievement demonstrates that ECS can manage specific morphological structures more effectively; however, limits persist in certain instances, such as over-stemming in the terms '*janlah*' and '*arokan*'.

Figure 4. Error Analysis *(Over-Stemming)*

3.4 Under-Stemming

The following table illustrates cases of under-stemming, where the algorithm does not fully strip the affixes from the original words. As a result, the output still contains remnants of prefixes or suffixes, failing to reach the correct root form. These incomplete reductions can lead to ambiguity or errors in downstream tasks such as translation and semantic interpretation.

Table 6. Example of Under-Stemming

Affix Word	Under-Stemming	Rootword	English
man-gana	ngana	kana	remember
ma-ningga-an	maningga	tingga	live

In natural language processing (NLP), under-stemming denotes a scenario in which the stemming procedure inadequately eliminates all affixes required to precisely identify the root word. Table 6 illustrates instances of this difficulty, showcasing the inadequacies of specific stemming algorithms in addressing intricate linguistic structures. For example, we should ideally reduce terms that begin with the prefix *'man-'*, like *'man-gana,'* to the root form *'kana'*, which in English means *'remember'*. Nonetheless, an erroneous stemming procedure yields *"ngana,"* which fails to accurately denote the intended root word. Similarly, the method misprocesses terms like *'maningga'* that feature the prefix *'ma-'* and the suffix *'-an'*. The method truncates the word '*tingga*', which signifies *'to live,'* resulting in an incomplete form instead of producing the accurate base form. This event highlights the intrinsic challenge of developing a reliable stemming system that can precisely identify root word forms in morphologically complex languages. The absence of stemming not only diminishes the precision of root word extraction but also has considerable repercussions for advanced tasks in NLP, including semantic analysis and syntactic parsing.

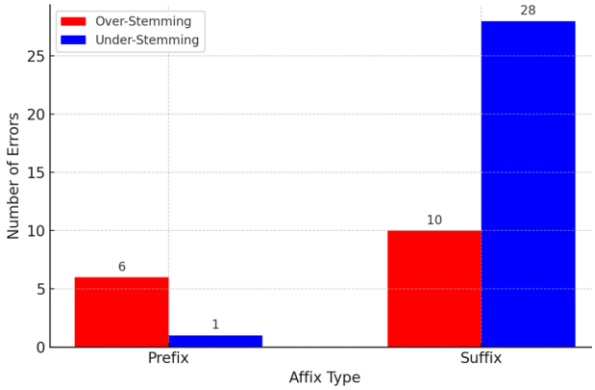

Figure 5. Error Distribution Over and Under-Stemming

The bar chart in Figure 5 illustrates the distribution of stemming errors based on affix types, specifically comparing over-stemming and under-stemming errors for prefixes and suffixes. On the x-axis, the chart categorizes errors into two affix types: prefix and suffix, while the y-axis shows the number of errors. The red bars represent the number of over-stemming errors, and the blue bars represent under-stemming errors. For prefixes, there are 6 instances of over-stemming errors compared to only 1 instance of under-stemming, indicating that over-stemming is more common with prefixes. In contrast, for suffixes, the number of under-stemming errors (28) far exceeds the number of over-stemming errors (10), suggesting that the algorithm tends to insufficiently remove suffixes more often than excessively removing them. This distribution highlights a significant challenge in the stemming process: the algorithm over-removes prefix affixes but under-removes suffix affixes. Understanding this error pattern is crucial for refining the stemming algorithm to improve its accuracy, particularly for languages with complex affixation such as Minangkabau.

3.5 Rules are Ineffective

The following table presents examples where rule-based stemming fails to produce the correct root words, despite the presence of recognizable affixes. These cases demonstrate that rigid affix-stripping rules may not always align with the morphological and semantic patterns of the language, resulting in inaccurate stems.

Table 7. Example of Rules Don't Work

Affix Word	After Stemming	Rootword	English
ma-iriang	mairiang	iriang	accompaniment
ma-mutuih	mamutuih	putuih	break up
ma-mukek	mamukek	pukek	noose
ma-nangguang	manangguang	tangguang	responsibility
ta-acuah-kan	taacuahkan	acuah	indifferent
ta-ubek	taubek	ubek	medicine
mang-icuah	mangicuah	kicuah	trick

Based on Table 7 regarding rules that do not work as intended, there were 687 words analysed during stemming procces, 130 couldn't be accurately reduced to their root forms. This issue stems from the incorrect application or misreading of particular rules intended to correspond with the intricate morphology of the Minangkabau language. These flaws underscore the intrinsic difficulties in developing a reliable stemming system that can consistently conform to language standards. These unprocessed words, identified as errors in accuracy testing, highlight the algorithm's limitations in managing specific morphological features. Table 10 illustrates specific examples of inaccurately executed flawed rules. For example, words adhering to the pattern (ma-V), such as *'ma-iriang'*, do not correctly diminish to their root form. Likewise, phrases such as *'ma-mutuih'* (root: *'putuih'*), *'mang-icuah'* (root: *'kicuah'*), (ta-V) *'ta-acuah-kan*' (root: *'acuah'*), and (ta-K) *'ta-ubek'* (root:*'ubek'*) exemplify the challenges faced. These examples show that we cannot routinely apply the fundamental rules of stemming across all linguistic contexts, especially in languages characterized by significant morphological complexity.

This highlights the complexity of formulating accurate and effective stemming rules for a language such as

Minangkabau, where various affixation patterns and distinct phonetic combinations pose considerable hurdles. To solve these problems, we need a more complex plan that includes rules that change depending on the situation, more language resources, and maybe even machine learning techniques to make stemming algorithms more flexible and accurate. These developments are crucial for attaining dependable text processing and facilitating wider applications in natural language processing for regional languages.

3.6 Comparison with other Methods

The following table summarizes the performance comparison of several stemming methods based on four key evaluation metrics: accuracy, precision, recall, and F1 score. As shown, the proposed method Modified ECS achieves the highest scores across all metrics, demonstrating its superior ability to reduce words to their correct root forms while minimizing stemming errors. This comparison highlights the effectiveness of refining rule-based approaches in addressing the morphological complexity of the Minangkabau language.

Table 8 compares the performance of four stemming methods based on accuracy, precision, recall, and F1 score. The Modified Enhanced Confix Stripping (ECS) method outperforms the others, achieving the highest accuracy (77.87%), precision (96.05%), recall (80.45%), and F1 score (87.56%). This demonstrates its superior ability to correctly identify and extract root words while minimizing errors. The Original ECS method ranks second, with slightly lower scores, indicating that the modifications made to ECS enhanced its effectiveness. The Nazief & Adriani method follows, showing reasonable performance but still falling behind the ECS-based approaches. Lastly, the Porter Stemming method performs the weakest across all metrics, suggesting it is less suitable for handling the morphological complexity of the language studied. Overall, the results emphasize that the Modified ECS algorithm is the most reliable and effective method for stemming in this context.

Table 8. Comparison of Stemming Methods Performance

Method	Accuracy	Precision	Recall	F1 Score
Modified ECS (Proposed Method)	77,87%	96,05%	80,45%	87,56%
Original ECS	75,13%	93,64%	78,86%	85,57%
Nazief & Adriani	73,10%	92,74%	77,29%	84,48%
Porter Stemming	68,89%	90,60%	72,73%	80,21%

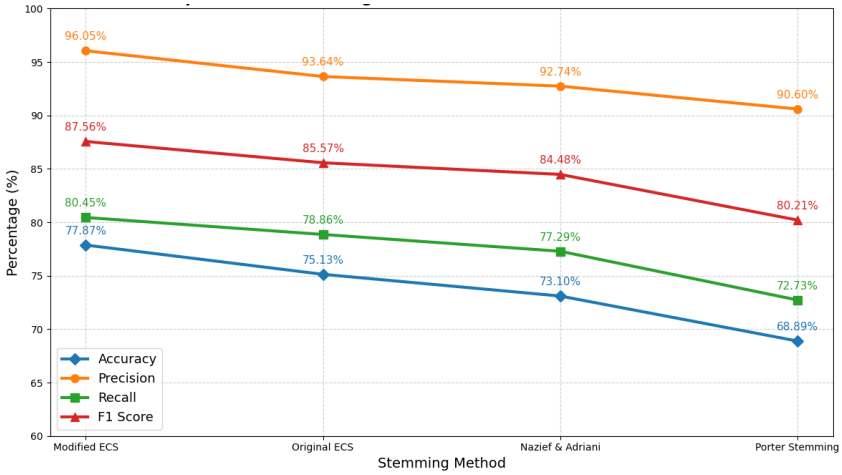

Figure 6. Performance Comparison of Stemming Methods

As shown in Figure 6, the results indicate that precision consistently achieves the highest values across all methods, suggesting that when a method identifies a word to be stemmed, it is very likely to be correct. Modified ECS stands out with the highest precision, approximately 96.05%. Both accuracy and recall exhibit a decreasing trend from Modified ECS to Porter Stemming, with Modified ECS achieving the highest accuracy (around 77.87%) and recall (80.45%), demonstrating its superior ability to correctly identify and process most stemmable words. The F1 score, which balances precision and recall, also places Modified ECS as the top performer, followed by Original ECS, Nazief & Adriani, and Porter Stemming. Notably, Porter Stemming shows the lowest performance across all metrics, underscoring its relative ineffectiveness compared to the other methods. Overall, the graph demonstrates that the Modified ECS method delivers the most accurate and balanced stemming performance, reinforcing the importance of refining stemming algorithms to enhance language processing—particularly for morphologically rich languages such as Minangkabau.

3.7 Evaluation and Accuracy

The evaluation of the stemming algorithm, based on the confusion matrix, shows that it correctly stemmed 535 words (true positives) and incorrectly stemmed 22 words (false positives), while failing to stem 130 words that should have been processed (false negatives). No

true negatives were recorded in this test. The overall accuracy is 77.87%, indicating that the algorithm correctly processed most words. Precision is notably high at 96.05%, meaning that when the algorithm identifies a word for stemming, it is very likely to be correct. Recall stands at 80.45%, reflecting the algorithm's ability to identify most words requiring stemming, although some were missed. The F1 score, as the harmonic mean of precision and recall, is 87.56%, demonstrating a good balance between correctly identifying stemmable words and minimizing false positives. These metrics indicate that while the algorithm is precise and effective, there remains room for improvement, especially in enhancing recall to reduce missed words.

The Cohen's Kappa statistic was calculated to assess the agreement between the stemming algorithm's predictions and the actual outcomes, beyond what would be expected by chance. The observed accuracy (Po) was approximately 77.87%, indicating that the algorithm correctly processed the majority of words. However, when accounting for the expected agreement (Pe), which considers the distribution of positive and negative cases in both actual and predicted labels, the value was about 78.95%. This unexpectedly high expected agreement resulted in a negative Cohen's Kappa value of approximately -0.0513. A negative Kappa suggests that the agreement between the algorithm's output and the true labels is worse than random chance, likely due to class imbalance or skewed prediction distributions. In this case, the very low number of true negatives combined with a high prevalence of positive cases affected the calculation, rendering Cohen's Kappa less informative. This result highlights the limitations of using Cohen's Kappa in highly imbalanced datasets and suggests that additional or alternative evaluation metrics may be necessary to fully capture the algorithm's performance.

4. Conclusions

The test results using the ECS stemming method revealed that out of 188 Minangkabau words with suffixes, 146 were accurately identified, while 28 were not effectively processed. Additionally, 10 words experienced over-stemming and 4 exhibited under-stemming, resulting in an average accuracy of 77.6%. For prefix-based Minangkabau words, 389 out of 499 were correctly stemmed, while 102 were inaccurately processed. In this category, 6 words showed over-stemming and 1 showed under-stemming, producing an average accuracy rate of 78.1%. Combining both evaluations, the overall average accuracy achieved was 77.8%. Despite these promising results, several limitations of the ECS stemming approach were observed, which highlight areas for future improvement. One of the main challenges is that stemming for regional languages such as Minangkabau must follow principles grounded in the language's unique morphological structure. The more affixes a word contains, the more complex the required rules become. Furthermore, the application of multiple rules across test samples often leads to overlapping conditions, which can result in excessive word reduction and thus over-stemming. Another significant limitation is the reliance on manual evaluation, which involves comparing the number of correctly stemmed words against errors including over-stemming, under-stemming, and failures of certain rules to produce the correct root form. This manual process is time-consuming and prevents the accuracy from reaching its optimal level. To overcome these issues, it is essential to develop a more robust methodology. One potential solution is to explore a hybrid approach that integrates rule-based stemming with machine translation techniques, specifically tailored for the complexities of regional language processing.

Acknowledgements

The authors extend their sincere gratitude to Universitas Negeri Malang for the internal research grant that supported the publication of this article as part of the doctoral study requirements.

References

[1] Z. Abidin, A. Junaidi, and Wamiliana, "Text Stemming and Lemmatization of Regional Languages in Indonesia: A Systematic Literature Review," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 10, no. 2, Art. no. 2, Jun. 2024, doi: 10.20473/jisebi.10.2.217-231.

[2] A. Arif siswandi, Y. Permana, and A. Emarilis, "Stemming Analysis Indonesian Language News Text with Porter Algorithm," *J. Phys. Conf. Ser.*, vol. 1845, no. 1, p. 012019, Mar. 2021, doi: 10.1088/1742-6596/1845/1/012019.

[3] M. E. Polus and T. Abbas, "Development for Performance of Porter Stemmer Algorithm," Feb. 26, 2021, *Social Science Research Network, Rochester, NY*: 3801021. Accessed: Mar. 24, 2025. [Online]. Available: https://papers.ssrn.com/abstract=3801021

[4] A. Jabbar, S. Iqbal, M. I. Tamimy, A. Rehman, S. A. Bahaj, and T. Saba, "An Analytical Analysis of Text Stemming Methodologies in Information Retrieval and Natural Language Processing Systems," *IEEE Access*, vol. 11, pp. 133681–133702, 2023, doi: 10.1109/ACCESS.2023.3332710.

[5] M. Alyousf and M. F. Alhalabi, "A Survey of Document Stemming Algorithms in Information Retrieval Systems," *ACM Trans Asian Low-Resour Lang Inf Process*, vol. 24, no. 4, p. 36:1-36:28, Mar. 2025, doi: 10.1145/3715120.

[6] S. Memon, G. Ali, K. N. M. -, A. Shaikh, S. K.Aasoori, and F. Ul, "Comparative Study of Truncating and Statistical Stemming Algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 2, 2020, doi: 10.14569/IJACSA.2020.0110272.

[7] Rianto, A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, "Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation," *J. Big Data*, vol. 8, no. 1, p. 26, Jan. 2021, doi: 10.1186/s40537-021-00413-1.

[8] H. Dwiharyono and S. Suyanto, "Stemming for Better Indonesian Text-to-Phoneme," *Ampersand*, vol. 9, p.

100083, Jan. 2022, doi: 10.1016/j.amper.2022.100083.
[9] H. A. Almuzaini and A. M. Azmi, "Impact of Stemming and Word Embedding on Deep Learning-Based Arabic Text Categorization," *IEEE Access*, vol. 8, pp. 127913–127928, 2020, doi: 10.1109/ACCESS.2020.3009217.
[10] J. Atwan, M. Wedyan, Q. Bsoul, A. Hammadeen, and R. Alturki, "The Use of Stemming in the Arabic Text and Its Impact on the Accuracy of Classification," *Sci. Program.*, vol. 2021, no. 1, p. 1367210, 2021, doi: 10.1155/2021/1367210.
[11] D. N. de Oliveira and L. H. de C. Merschmann, "Joint evaluation of preprocessing tasks with classifiers for sentiment analysis in Brazilian Portuguese language," *Multimed. Tools Appl.*, vol. 80, no. 10, pp. 15391–15412, Apr. 2021, doi: 10.1007/s11042-020-10323-8.
[12] Department of Computer Science, Faculty of Mathematics and Informatics, University Mohamed Boudiaf of M'sila, M'sila, Algeria, S. Gadri, E. Neuhold, and Department of Computer Science, Faculty of Computer Science, University of Vienna, Vienna, Austria, "Developing a Multilingual Stemmer for the Requirement of Text Categorization and Information Retrieval," *Int. J. Electr. Eng. Inform.*, vol. 14, no. 2, pp. 291–310, Jun. 2022, doi: 10.15676/ijeei.2022.14.2.3.
[13] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimed. Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.
[14] C. Nnaji and A. A. Karakhan, "Technologies for safety and health management in construction: Current use, implementation benefits and limitations, and adoption barriers," *J. Build. Eng.*, vol. 29, p. 101212, May 2020, doi: 10.1016/j.jobe.2020.101212.
[15] B. Priya Kamath *et al.*, "Comprehensive Analysis of Word Embedding Models and Design of Effective Feature Vector for Classification of Amazon Product Reviews," *IEEE Access*, vol. 13, pp. 25239–25255, 2025, doi: 10.1109/ACCESS.2025.3536631.
[16] Rianto, A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, "Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation," *J. Big Data*, vol. 8, no. 1, p. 26, Jan. 2021, doi: 10.1186/s40537-021-00413-1.
[17] A. Jabbar, S. Iqbal, M. I. Tamimy, A. Rehman, S. A. Bahaj, and T. Saba, "An Analytical Analysis of Text Stemming Methodologies in Information Retrieval and Natural Language Processing Systems," *IEEE Access*, vol. 11, pp. 133681–133702, 2023, doi: 10.1109/ACCESS.2023.3332710.
[18] C. Wu *et al.*, "Natural language processing for smart construction: Current status and future directions," *Autom. Constr.*, vol. 134, p. 104059, Feb. 2022, doi: 10.1016/j.autcon.2021.104059.
[19] S. Kulkarni and S. F. Rodd, "Context Aware Recommendation Systems: A review of the state of the art techniques," *Comput. Sci. Rev.*, vol. 37, p. 100255, Aug. 2020, doi: 10.1016/j.cosrev.2020.100255.
[20] A. A. Afifi and F. Yufriadi, "THE COEXISTENCE OF KAUM MUDO AND KAUM TUO: THE TRANSFORMATION OF ISLAMIC EDUCATION IN MINANGKABAU," 2024.
[21] H. H. Park, K. J. Zhang, C. Haley, K. Steimel, H. Liu, and L. Schwartz, "Morphology Matters: A Multilingual Language Modeling Analysis", doi: 10.1162/tacl_a_00365.
[22] S. Santuso and S. Sukarno, "Contrastive analysis of form and meaning of reduplication in Madurese and Minangkabau language," *J. Lang. Lit. Soc. Cult. Stud.*, vol. 3, no. 1, Art. no. 1, Mar. 2025, doi: 10.58881/jllscs.v3i1.276.
[23] S. Jatmika, S. Patmanthara, A. P. Wibawa, and F. Kurniawan, "Cognition-Based Document Matching Within the Chatbot Modeling Framework," *J. Appl. Data Sci.*, vol. 5, no. 2, Art. no. 2, May 2024, doi: 10.47738/jads.v5i2.209.
[24] S. Supriyono, A. P. Wibawa, S. Suyono, and F. Kurniawan, "Analyzing Audience Sentiments in Digital Comedy: A Study of YouTube Comments Using LSTM Models," *J. Appl. Data Sci.*, vol. 5, no. 4, Art. no. 4, Oct. 2024, doi: 10.47738/jads.v5i4.393.
[25] D. W. Otter, J. R. Medina, and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 604–624, Feb. 2021, doi: 10.1109/TNNLS.2020.2979670.
[26] H. Handoko, "Developing the Corpus of Minangkabau Language: Insights, Challenges, and Future Directions," *J. ARBITRER*, vol. 11, no. 3, Art. no. 3, Sep. 2024, doi: 10.25077/ar.11.3.413-429.2024.
[27] Z. Abidin, A. Junaidi, and Wamiliana, "Text Stemming and Lemmatization of Regional Languages in Indonesia: A Systematic Literature Review," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 10, no. 2, Art. no. 2, Jun. 2024, doi: 10.20473/jisebi.10.2.217-231.
[28] N. Pittaras, G. Giannakopoulos, G. Papadakis, and V. Karkaletsis, "Text classification with semantically enriched word embeddings," *Nat. Lang. Eng.*, vol. 27, no. 4, pp. 391–425, Jul. 2021, doi: 10.1017/S1351324920000170.
[29] Z. Abidin, A. Junaidi, and Wamiliana, "Text Stemming and Lemmatization of Regional Languages in Indonesia: A Systematic Literature Review," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 10, no. 2, Art. no. 2, Jun. 2024, doi: 10.20473/jisebi.10.2.217-231.
[30] V. P. Carolina, E. Utami, and A. Yaqin, "Exploring Stemming Techniques in Ambon Malay Languages: A Systematic Literature Review," *Jambura J. Inform.*, vol. 6, no. 1, Art. no. 1, May 2024, doi: 10.37905/jji.v6i1.24954.
[31] A. Mardatillah, "The enterprise culture heritage of Minangkabau cuisine, West Sumatra of Indonesia as a source of sustainable competitive advantage," *J. Ethn. Foods*, vol. 7, no. 1, p. 34, Sep. 2020, doi: 10.1186/s42779-020-00059-z.
[32] M. Irhamni and K. Nasution, "Prefix in Grammatical Meaning of Minang Kabau Language," *J. Ilm. Wahana Pendidik.*, vol. 10, no. 12, Art. no. 12, Jun. 2024, doi: 10.5281/zenodo.12542283.
[33] S. I. Melia, J. Sholihah, D. Nisak, I. S. Juniaristha, and A. T. Ni'mah, "The Ngoko Javanese Stemmer uses the Enhanced Confix Stripping Stemmer Method," *Rekayasa*, vol. 16, no. 1, Art. no. 1, Apr. 2023, doi: 10.21107/rekayasa.v16i1.19308.
[34] N. W. Wardani and P. G. S. C. Nugraha, "Stemming Teks Bahasa Bali dengan Algoritma Enhanced Confix Stripping," *Int. J. Nat. Sci. Eng.*, vol. 4, no. 3, Art. no. 3, Dec. 2020, doi: 10.23887/ijnse.v4i3.30309.

[35] V. P. Carolina, E. Utami, and A. Yaqin, "Exploring Stemming Techniques in Ambon Malay Languages: A Systematic Literature Review," *Jambura J. Inform.*, vol. 6, no. 1, Art. no. 1, May 2024, doi: 10.37905/jji.v6i1.24954.

[36] D. Soyusiawaty, A. H. S. Jones, and N. L. Lestariw, "The Stemming Application on Affixed Javanese Words by using Nazief and Adriani Algorithm," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 771, no. 1, p. 012026, Mar. 2020, doi: 10.1088/1757-899X/771/1/012026.

[37] D. A. Sulistyo, A. P. Wibawa, D. D. Prasetya, and F. A. Ahda, "LSTM-Based Machine Translation for Madurese-Indonesian," *J. Appl. Data Sci.*, vol. 4, no. 3, Art. no. 3, Sep. 2023, doi: 10.47738/jads.v4i3.113.

[38] D. A. Sulistyo, A. P. Wibawa, D. D. Prasetya, and F. A. Ahda, "An enhanced pivot-based neural machine translation for low-resource languages," *Int. J. Adv. Intell. Inform.*, vol. 11, no. 2, Art. no. 2, May 2025, doi: 10.26555/ijain.v11i2.2115.

[39] F. A. Ahda, A. P. Wibawa, D. D. Prasetya, and D. A. Sulistyo, "Comparison of Adam Optimization and RMS prop in Minangkabau-Indonesian Bidirectional Translation with Neural Machine Translation," *JOIV Int. J. Inform. Vis.*, vol. 8, no. 1, pp. 231–238, Mar. 2024, doi: 10.62527/joiv.8.1.1818.

[40] H. Handoko, "Developing the Corpus of Minangkabau Language: Insights, Challenges, and Future Directions," *J. ARBITRER*, vol. 11, no. 3, pp. 413–429, Sep. 2024, doi: 10.25077/ar.11.3.413-429.2024.

[41] S. Z. Fazekas, R. Mercaş, and D. Reidenbach, "On the Prefix–Suffix Duplication Reduction," *Int. J. Found. Comput. Sci.*, vol. 31, no. 01, pp. 91–102, Jan. 2020, doi: 10.1142/S0129054120400067.

[42] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimed. Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.

[43] A. Theissler, M. Thomas, M. Burch, and F. Gerschner, "ConfusionVis: Comparative evaluation and selection of multi-class classifiers based on confusion matrices," *Knowl.-Based Syst.*, vol. 247, p. 108651, Jul. 2022, doi: 10.1016/j.knosys.2022.108651.

[44] L. Vergni, F. Todisco, and B. Di Lena, "Evaluation of the similarity between drought indices by correlation analysis and Cohen's Kappa test in a Mediterranean area," *Nat. Hazards*, vol. 108, no. 2, pp. 2187–2209, Sep. 2021, doi: 10.1007/s11069-021-04775-w.