JURNAL INFOTEL
Informatics - Telecommunication - Electronics
Website: http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN: 2085-3688; e-ISSN: 2460-0997

Time complexity in rejang language stemming

Sastya Hendri Wibowo[1], Rozali Toyib[2,*], Muntahanah[3], Yulia Darnita[4]
[1-4]Universitas Muhammadiyah Bengkulu
[1-4]Jl. Bali, Bengkulu 38119, Indonesia
[*]Corresponding email: rozalitoyib@umb.ac.id

Received 5 March 2022, Revised 28 June 2022, Accepted 30 June 2022

Abstract — The Rejang language belongs to the Malay language family, which is almost spread throughout Indonesia and has its uniqueness that is different from Malay in general. The speakers spread across six districts, namely Rejang Lebong Regency, Lebong Regency, Kepahiang Regency, North Bengkulu Regency, Central Bengkulu, and Bengkulu City, the majority are used in Bengkulu Province. Rejang language has a complex and complicated word structure so that it is easily understood by people, so a language dictionary is made in the form of a printed book or digital dictionary that allows everyone to access it anywhere and anytime, in Bengkulu currently there is no digital one. dictionary, while to build a language translator application a stemming algorithm is needed. is the process of separating basic words from affixed words in sentences by separating basic words and affixes which can consist of prefixes, infixes, and, suffixes. The application of the Stemming Algorithm in the Rejang language for the preservation of local culture so as not to lose its cultural roots and the specific purpose of designing a model for implementing the Rejang stemming algorithm in the Rejang language in the form of an AI-integrated database system. The algorithms used are Enhanced Confix Striping (ECS), New Enhances Confix Striping (NECS,) and Rejang Algorithm where the affix groups in ECS and NECS make modifications to the prefixes which are grouped into several groups of prefixes, suffix, and affixes are grouped into several groups. suffix group. The results of this study have succeeded in building an efficient and effective Rejang language stemming algorithm, where efficiency is indicated by the time complexity of the $O(\log n)$ algorithm, and the effectiveness is shown from the results of accuracy of 99% against the 9000-word test with affixes, this accuracy value indicates that there has been over stemming and under stemming 1% with a failure rate of 1% from 15 documents tested and the average speed of complexity is between 4.27 seconds to 111.754 seconds.

Keywords – rejang, time complexity, stemming

I. Introduction

Each province in Indonesia has its regional dialect, such as Sundanese from West Java Province, Minang from West Sumatra Province, Batak from North Sumatra Province, and Rejang from Bengkulu Province. Although most provinces have local language libraries in the form of books, they do not have digital libraries, Building a digital dictionary is the main requirement as a source of supporting data used by language translator applications.

In Bengkulu, there is currently no Rejang digital dictionary for the Rejang language translator application to Indonesian, while to build a language translator application a stemming algorithm is needed. Stemming is the process of separating the basic word from an affixed word in a sentence by separating the base word and affixes which can consist of prefixes (prefixes), insertions (infixes), and suffixes (suffixes) [1]. Every language with other languages generally has differences due to differences in morphology, such as the Javanese language has a different morphology from the Rejang language. The stemming algorithm is influenced by several factors, including Overstemming, Understemming, Unchanged, and Spelling exceptions [2]. The fastest stemming process is found in the Vega Algorithm, the highest accuracy is found in the Nazief and Adriani algorithms [3], [4].

Rejang language is a local language in the Rejang tribal community located in Bengkulu Province with speakers spread over four districts, namely Rejang Lebong Regency, Lebong Regency, Kepahiang Regency, and North Bengkulu Regency [4]. At least five languages are recorded in the Bengkulu Province: the Bengkulu language, Enggano language,

https://doi.org/10.20895/infotel.v14i3.764

Rejang language, Javanese language, and Sundanese language [5]. The algorithm used to improve the performance of Information Retrieval (IR) is a process of separating documents that are considered relevant from a set of available documents [6]–[8].

The stemming process ultimately increases the number of documents retrieved in the IR system. Text grouping, categorization, and summarization also require these conversions as part of the pre-processing before actually implementing the related algorithms [9], [10].

The proposed algorithm is applied to 41 different sets of QAPLIB benchmarks, while the obtained results show that the proposed algorithm is more efficient and accurate compared to Taboo Search, Evolutionary Differential, and Genetics algorithms. both and convex relaxation achieve good sparse coefficients as compared to other optimization algorithms. The resulting BP is more accurate than the greedy technique with a slight increase in processing time [11], [12].

The Stemmer method by using the morphology of a word has several drawbacks, such as inappropriately removing the prefixes on basic words starting with the letters ”k”, ”t”, ”s”, and ”p” and inappropriate in removing suffixes, especially for the suffixes ”-kan” and ”- an.” [13]–[18].

The morphology of the Indonesian language is not only based on the data of teaching materials that have been used so far but also new teaching materials are sought according to the needs of students, including those related to the Jambi Malay morphophonemic process as part of local wisdom as demanded by the vision and mission of the Study Program. First, we have to understand the process of forming words and morphemes as the smallest part of the language itself, Misguided in Indonesian that keeps happening is of course not good for the development of Indonesian, this misunderstanding in Indonesian will be avoided if language users want to learn the rules of the language [19]–[24].

II. Research Method

The Rejang stemming algorithm was developed by first studying the workings of the Enhanced Confix Striping (ECS), New Enhances Confix Striping (NECS and stemming algorithms. In this study, the researchers used the stemming algorithm as the main reference in making the Rejang algorithm. Next, analyze the strengths and weaknesses of the three algorithms. After studying how it works and analyzing the strengths and weaknesses of the Indonesian stemming algorithm as a basis for reference or guidelines, a stemming algorithm for the Rejang language was made based on the morphology of the Rejang language.

A. Building a Rejang Language Digital Dictionary

The steps for making a digital dictionary are as follows.

1) Inventory or Data Collection Rejang Language Basic Words

The basic words used in this study were taken from the Rejang language Dictionary published by the Bengkulu Provincial Language Office of the Ministry of Education and Culture in 2013 which consists of 6983 Rejang language basic words.

2) Database Creation of Rejang Language Basic Word Dictionary

The database structure of the Rejang language base word consists of 3 fields, namely Id which is the primary key, Base-Word which contains the Rejang language basic word, and Meaning which contains the meaning of the Rejang language word in Indonesian. The database structure of the Rejang language dictionary is shown in Table 1.

Table 1. Database Structure of Rejang Language Dictionary

Table Name	Field Name	Data Type	Char. Length	Description
Katadasar_ rejang	Id	Int	5	Primary key
	Kata_ dasar	Varchar	50	
	Arti	Varchar	50	

B. Study and Analysis of Rejang Language Morphology

To create a new stemming algorithm, the Rejang Algorithm, we first begin by analyzing the groups, sequences, and methods of deleting the morphological affixes of the Rejang language.

1) Analysis of Rejang Language Structure

The Rejang language belongs to the Malay or Indonesian language family. The syntax structure follows the DM law, the element being explained (D) is followed by an explanatory element (M), for example, umeak lei ”big house”: umeak, as the element being explained (D) followed by lei, as the element explaining (M). Tabel 2 is an example of the application of the DM law in a verb phrase.

Table 2. Example of the Application of the DM Law in Verb Phrases

Explained	Explain
Temot ’Duduk’	*Temenung ’Termenung’*
Bekerjo ’Bekerja’	*Tesatuk ’Sempurna’*
Majak ’Mengajak’	*Besaweak ’Bersawah’*
Melilei ’Berlari’	*Terus ’Terus’*
Bedan ’Berhenti’	*Kileak ’Dahulu’*

The example above also shows that the verb Rejang language has its system. In this system some verbs consist of one morpheme, for example, temot ’sit’, Bedan ’stop’, and verbs that consist of two morphemes, for example, bekerjo ’work’, majak ’to invite', melilei 'barlari', besaweak 'besawah'. Verbs consist of two morphemes, namely bound morpheme and free morpheme. The bound morpheme in the example above is a prefix, namely bekerjo and besaweak, men- in majak and me- lilei. In addition, in the example above there is an original form, namely – invite in majak. Even though it has started the process, the original form cannot be used as a free morpheme in syntactic construction. The free morphemes in the example above are temot and Bedan.

2) Analysis of the Rejang Language Affix Group

Affixes in the Rejang language consist of prefixes, infixes, and suffixes, in the Rejang language there are no affixes called confixes, there are combined affixes, as shown in Table 3.

Table 3. Groups of Morphological Affixes for Rejang

No.	Affix	Group
1	Prefix	*”men-”,”ke-”,”be-”,”ne-”,”te”-,”se-”*
2	Insert	*”-em-”,”-en-”*
3	Suffix	*”-ke”*
4	Combined (prefix and prefix)	*”be-”+”ke-”* *”te-”+”ke-”* *”se-”+”ke-”*
5	Combined (prefix and suffix)	*”ke-”+”-ke”*

3) Morphological Process Analysis

The morphological process of the Rejang language is the process of adding affixes to the basic words of the Rejang language. Affixes in Rejang language consist of: prefix group (men-, ke-, be-, ne-, te, se), insertion group (-em-, -en-), suffix group (-ke), prefix compound group - prefix (be- + ke-, te- + ke-, se- + ke-), and prefix-suffix compound groups (ke- + -ke). The addition of affixes can cause conditions: 1) changes in phonemes (vocals) contained in basic words, 2) changes in the form of basic words, and 3) no changes in form in basic words.

a) Analysis of the addition of affixes that cause changes in the form of the basic word, consisting of:

1) Prefix Me-, Meng-, Men-
2) Added Insert -Em-
3) Adding Inserts -En-

b) Analysis of the addition of affixes that do not change the form of the basic word:

1) Adding the Prefix Men-
2) Adding Prefix To-
3) Adding the Prefix Be-
4) Adding the Prefix Ne-
5) Adding the Prefix Te-
6) Adding the Prefix Se-
7) Addition of the suffix to-

C. Building the Rejang Language Stemming Algorithm Based on the Results of the Rejang Language Morphology Analysis

1) Deletion Sequence Creation

In the development of the Rejang Language stemming algorithm, the idea of sorting for deletion of affixes begins with the deletion of prefixes, followed by deletion of insertions, then deletion of suffixes, and finally deletion of combinations.

2) Sequencing the Deletion Process in Detail

The Rejang algorithm is an algorithm based on the grouping of morphological affixes of the Rejang language. The order and method of deleting affixes in the Rejang language stemming algorithm are expressed by a sequence of affix deletion rules consisting of prefixes, insertions, and suffixes. The process of removing affixes includes the prefix ”men”, ”ke”, ”be”, ”ne”, ”te”, ”se”, insertion of ”em”, ”en”, and the suffix ”ke”.

D. Analyzing Algorithm Performance

The Rejang language stemming algorithm is an algorithm built from the searching procedure to find basic words in the Rejang language digital dictionary. The searching procedure used in this research is the binary searching procedure which has a time complexity of $O(\log n)$, where n is the number of basic words in the Rejang language digital dictionary. The main procedure consists of procedure 1 (check prefix), procedure 2 (check insertion), procedure 3 (check suffix), and procedure 4 (check merge). In each sub-procedure, each path in it all uses the searching procedure. Then the time complexity of the main procedure is the MAXIMUM value of the time complexity of the delete procedure 1, 2, 3, or 4, *i.e.*, MAXIMUM [4 $O(\log n)$, $O(\log n)$, $O(\log n)$, $O(\log n)$] = 4 $O(\log n)O(\log n)$.

The complexity of the algorithm (main procedure) of 4 $O(\log n)$ is obtained from the analysis of the time complexity calculation of the elimination procedures 1, 2, 3, and 4 which are described as follows.

Subprocedure 1 (check prefix) consists of procedure 1A (check prefix men), procedure 1B (check prefix to), procedure 1C (check prefix be), procedure 1D (check prefix te), and procedure 1F (check prefix to). . In each sub-procedure, each path in it all uses the searching procedure. Then the time complexity of procedure 1 is the sum of the time complexity of the MAXIMUM value of the time complexity of procedures 1A, 1B, 1C, 1D, 1E, and 1F which is MAXIMUM [4 $O(\log n)$, $O(\log n)$, $O(\log n)$, $O(\log n)$, $O(\log n)$, $O(\log n)$] = 4 $O(\log n)$.

Subprocedure 2 (Check insert) consists of procedure 2A (check em insertion), and procedure 2B (check en insertion). In each sub-procedure, each path in it all uses the searching procedure. Then the time complexity of procedure 2 is the maximum value of the time complexity of 2A and 2B, namely MAXIMUM [$O(\log n)$, $O(\log n)$] = $O(\log n)$.

In subprocedure 4 (check joint) consists of procedure 4A (check joint Keke), procedure 4B (check joint teke), procedure 4C (check joint Keke),

Table 4. Results of Testing Lines of Words in a Document

No.	File Name	Number of Words	Number of Detected	Affixes Processing Time per Second	Number of Words with Unsuccessful Affixes in Stemming
1	Bma2.txt	97	14	5.087	2
2	Bma3.txt	109	18	4.2722	0
3	Bma4.txt	206	20	8.1632	1
4	Bma5.txt	106	8	4.1578	0
5	Buku1.txt	609	34	21.053	1
6	Buku2.txt	340	21	11.155	2
7	Buku3.txt	368	29	37.0117	0
8	Buku4.txt	556	57	33.4877	6
9	Buku5.txt	374	31	66.2444	3
10	Buku6.txt	263	12	10.1945	1
11	Buku7.txt	376	28	59.9278	0
12	Buku8.txt	373	26	12.4483	0
13	Buku9.txt	326	29	111.7543	0
14	Buku10.txt	221	19	7.9445	2
15	Kejai1.txt	167	14	8.2316	2

and procedure 4D (check joint Keke). In each subprocedure, each path in it all uses the searching procedure. Then the time complexity of procedure 4 is the maximum value of the time complexity of 4A, 4B, 4C, and 4D *i.e.,* MAXIMUM [$O(\log n)$, $O(\log n)$, $O(\log n)$, $O(\log n)$] = $O(\log n)$.

In each subprocedure, each path in it all uses the searching procedure. So the time complexity of procedure 3 is only from the time complexity of the procedure for the Removal of Vowel Prefixes or Consonant Prefixes. Thus the time complexity of procedure 3 is $O(\log n)$.

III. RESULT

A. *Testing the Rejang Language Stemming Algorithm*

1) *Testing Affixes*

The Rejang digital dictionary contains 6983 basic words. Algorithm testing in this study was carried out on 9000 affixed words built from 1500 basic words of the Rejang language. The selection of the 1500 basic words is based on groups of basic words starting with vowels, starting with consonants, and starting with the letter 'k' where each of the 500 words is taken entirely from the digital dictionary of the Rejang language.

a) List of Affixed Words tested: The number of affixed words tested in the Rejang algorithm application was 9000, consisting of 18 groups of affixes with each group consisting of 500 words. Each group can consist of affixed words containing basic words starting with a vowel, and/or starting with a consonant, or starting with the letter "k".

b) Results of Testing Affixed Words: Stemming test results Rejang algorithm for multiple words affixed per affix group, test results 9000 suffix words in Rejang algorithm application (Success = 8973; Fail = 27).

2) *Testing a Document Containing a Line of Words*

The word line test by the REJANG Stemmer application was carried out on 15 documents, where the documents were taken from the Rejang language book: 1. Titled "Kelpiak Ukum Adat Ngen Riyan Ca'o Kutai Jang, Rejang Lebong Regency" 2. Titled "Ireak Ca 'o Kutei Jang" 3. Titled "Kejai Ca'o Malim". The book that was used as a test was initially still in the form of a printed book. Furthermore, to be used as test data on the Rejang Algorithm, the books are converted into digital books in .txt format. The output of the test results of a document are 1) the total number of words in the document, 2) the number of words with stemming affixes, and 3) the processing time for stemming length. The output of the test results of 15 documents is shown in Table 4.

The affixed words that do not change after stemming can be shown in Table 5.

Table 5. Affixed Words that Don't Change After Stemming in Documents

No.	Document Name	Suffix
1	Bma2.txt	Pemejem Ketengke
2	Bma4.txt	Temanem
3	Buku1.txt	Temulung
4	Buku2.txt	Benekemoa Benelaten
5	Buku4.txt	Keleake Kelemke Kembaske Kemboke Ketengke Keike
6	Buku5.txt	Senengak Temulung Kepenke
7	Buku6.txt	Temanem
8	Buku10.txt	Pemejem Temanem Benelaten Kepenke

B. *Comparison of Stemming Algorithms*

The results of the Rejang algorithm research in this article are a novelty in several respects compared to the Enhanced Confix Stripping (ECS) and New Enhanced Confix Stripping (NECS) algorithms.

1) *Affix Group*

The affix group of the Enhanced Confix Stripping (ECS) algorithm, New Enhanced Confix Stripping (NECS) modifies the prefixes which are grouped into several groups of prefixes, the suffixes are grouped into several groups of suffixes. While there is no insertion affix grouping. In the Rejang algorithm, the grouping of affixes is based on the morphology of the Rejang language, and there is no modification of the affix groups. A comparison of affix groups of each algorithm is shown in Table 6.

Table 6. ECS, NECS, and Rejang Algorithm Affix Groups

ECS Algorithm	**NECS Algorithm**	**Rejang**
Prefix Standard: di-, ke-, se-	Prefix 1: be, di, ke, me, pe, se, te, bel, ber, mem, men, pel, pem, pen, per, ter, kese, sepe berke,	Prefix: men, ke, be, ne, te, se
Prefix Complex: me-, be-, pe, te-	Prefix 2: meng, peng, meny, peny, mem, pem, mem, pen	be, pe, te
-	Prefix 3: anti, eks, ekstra, hiper, infra, makro, maha, ini	diter
Infix: -em-, -el-, -er-	Infix: er, em, el, aj, in	infix: em, en
Suffix: -kan, -an, -i	Suffix 1: kan	Suffix: ke
-	Suffix 2: I, isasi, an, er, wi, in, iah, wan, mun, logi, isme, onal, itas, is	-
Particle: -lah, -kah, -tah, -pun	Particle: lah, kah, tah, pun	-

2) Removal Order

The order of removal of affixes from the Enhanced Confix Stripping (ECS) algorithm is starting with particle removal, pronoun removal, suffix removal, standard prefix removal and complex prefix removal, New Enhanced Confix Stripping (NECS) algorithm begins with prefix 1 removal, prefix removal 2, deletion prefix 3, particle deletion, deletion of property suffix, deletion of suffix 1, deletion of suffix 2 and deletion of insertions, beginning with particle deletion, pronoun deletion, unprocessed prefix deletion, combined prefix deletion, morphophonemic deletion of prefixes and suffix deletion. As for the Rejang algorithm, it begins with the removal of the prefix, the removal of the insert, the removal of the suffix, and the removal of the merge. The order of deletion is shown in Table 7 below.

Table 7. Deletion Sequence of ECS, NECS, and Rejang Algorithms

ECS Algorithm	**NECS Algorithm**	**Rejang**
Delete Particle	Delete Prefix #1	1. Delete Prefix
Delete Change a word	Delete Prefix #2	Delete Insert
Delete Suffix	Delete Prefix #3	Delete Suffix
Delete Prefix Standard	Delete Particle	Delete Combined
Delete Prefix Particle	Delete Suffix	-
-	Delete Suffix 1	-
-	Delete Suffix n 2	-
-	Delete Infix	-

IV. Discussion

Rejang language digital dictionary database with a base word count of 7000 words, of which 1500 words are used to construct 9000 affixed words used in testing the Rejang language stemming algorithm.

From the results of analyzing the morphology of the process of adding affixes, it was found that 18 groups/types of the smallest affixes in the Rejang language were used as conditional statements in the Rejang language stemming from the Enhanced Confix Striping (ECS) Algorithm and New Enhances Confix Striping (NECS) Algorithm and the Test Results of 9000 Words Affixes in the Rejang Algorithm Application (Success= 8973; Fail= 27), in the process of adding affixes, it is determined that the process of deleting affixes in the Rejang language stemming Algorithm is in the sequence starting from the deletion of the prefix, then the deletion of the insertion, then the deletion of the suffix and finally the deletion of the merge.

V. Conclusion

This research has succeeded in building Rejang language stemming with the ECS Algorithm and NECS Algorithm which is efficient and effective, where efficiency is indicated by the time complexity of the O(log n) algorithm, and effectiveness is shown from the results, 99% accuracy against the 9000 words affix test. In other words, this accuracy value indicates that 1% has occurred over stemming and under stemming and this study also succeeded in testing 15 text documents with an average stemming failure rate of 1% with an average complexity time of 4.277 seconds to 111.7543 seconds.

References

[1] R. Setiawan, A. Kurniawan, W. Budiharto, I. H. Kartowisastro, and H. Prabowo, ”Flexible affix classification for stemming Indonesian Language,” in *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Jun. 26 – Jul. 1, Chiang Mai, Thailand, 2016. doi: 10.1109/ECTICon.2016.7561257.

[2] D. Wahyudi, T. Susyanto, and D. Nugroho, “Implementasi dan analisis algoritma stemming nazief & adriani dan porter pada dokumen berbahasa indonesia,” *KJurnal Ilmiah SINUS*, vol. 15, no. 2, pp. 49–56, 2017. doi: 10.30646/sinus.v15i2.305.

[3] M. S. H. Simarangkir, ”Studi perbandingan algoritma - algoritma stemming untuk dokumen teks bahasa indonesia,” *Jurnal Inkofar*, vol. 1, no. 1, pp. 40–46, 2017. doi: 10.46846/jurnalinkofar.v1i1.2.

[4] A. A. Magriyanti, ”Analisis pengembangan algoritma porter stemming dalam bahasa indonesia,” 2018. doi: 10.31227/osf.io/7ge4v.

[5] S. F. Wibowo, ”Fonem segmental dan distribusinya dalam bahasa rejang dialek musi,” *Jurnal Madah*, vol. 7, no. 1, pp. 37–46, 2016. doi: 10.31503/madah.v7i1.113.

[6] D. Muliadi, ”Perbandingan algoritma stemming untuk bahasa indonesia dengan parameter akurasi dan waktu proses,” pp. 7–37, 2015

[7] S. Wibowo, B. Soerowirdjo, Ernastuti, and A. Tarigan, ”Development of stemming algorithm for Rejang language stemmer based on Rejang language morphology,” *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 5 (Special Issue), pp. 1858–1870, 2019.

[8] A. Sinaga, Adiwijaya, and H. Nugroho, ”Development of word-based text compression algorithm for Indonesian language document,” in *2015 3rd Int. Conf. Inf. Commun. Technol. ICoICT 2015*, May 27–29, Bali, Indonesia, 2015, pp. 450–454, 2015. doi: 10.1109/ICoICT.2015.7231466.

[9] M. Watzlawik and J. Valsiner, ”The making of magic: cultural constructions of the mundane supernatural,” The Oxford handbook of culture and psychology, vol. 2, no. 6, pp. 1930–1938, 2012. doi: 10.1093/oxford he/9780195396430.013.0038.

[10] J. Wibowo, ”Aplikasi penentuan kata dasar dari kata berimbuhan pada kalimat bahasa indonesia dengan algoritma stemming,” *Jurnal Riset Komputer (JURIKOM)*, vol. 3, no. 5, pp. 346–350, 2016. doi: 10.30865/jurikom.v3i5.70.

[11] A. S. Hameed, B. M. Aboobaider, N. H. Choon, and M. L. Mutar, ”Improved discrete differential evolution algorithm in solving quadratic assignment problem for best solutions,” *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 9, no. 12, pp. 434–439, 2018. doi: 10.14569/IJACSA.2018.091261.

[12] A. Qayyum, A. S. Malik, M. Naufal, M. Mazher, R. F. Ahmad, and M. F. Abdullah, ”Evaluation of optimization algorithms for sparse and redundant dictionaries,” in *2015 IEEE Student Symposium in Biomedical Engineering & Sciences (ISSBES)*, Nov. 4, Shah Alam, Malaysia, 2015, pp. 128–133. doi: 10.1109/ISSBES.2015.7435882.

[13] P. B. S. Wiguna and B. S. Hantono, ”Peningkatan algoritma porter stemmer bahasa indonesia berdasarkan metode morfologi dengan mengaplikasikan 2 tingkat morfologi dan aturan kombinasi awalan dan akhiran,” *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 2, no. 2, pp. 1–6, 2013.

[14] T. Winarti, J. Kerami, and S. Arief, ”Determining term on text document clustering using algorithm of enhanced confix stripping stemming,” *International Journal of Computer Applications*, vol. 157, no. 9, pp. 8–13, 2017. doi: 10.5120/ijca2017912761.

[15] A. Purwarianti, ”A nondeterministic indonesian stemmer,” in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, Jul. 17–19, Bandung, Indonesia, 2011. doi: 10.1109/ICEEI.2011.6021829.

[16] A. Z. Arifin, I. P. A. K. Mahendra, and H. T. Ciptaningtyas, ”Enhanced confix stripping stemmer and ants algorithm for classifying news document in representation of textual,” in *The 5th International Conference on Information & Communication Technology and Systems*, 2009, pp. 149–158.

[17] Z. Abidin, A. Wijaya, and D. Pasha, ”Aplikasi stemming kata bahasa lampung dialek api menggunakan pendekatan brute-force dan pemograman C#,” *Jurnal Media Informatika Budidarma*, vol. 5, no. 1, p. 1–8, 2021. doi: 10.30865/mib.v5i1.2483.

[18] A. F. Zulfikar, ”Pengembangan algoritma stemming bahasa indonesia dengan pendekatan dictionary base stemming untuk menentukan kata dasar dari kata yang berimbuhan,” *Jurnal Informatika Universitas Pamulang*, vol. 2, no. 3, pp. 143–146, 2017. doi: 10.32493/informatika.v2i3.1443.

[19] Akhyaruddin, A. Agusti, and H. Yusra, ”Analisis morfofonemik bahasa melayu jambi sebagai pengembangan bahan ajar morfologi bahasa indonesia,” *Pena: Jurnal Pendidikan Bahasa dan Sastra*, vol. 9, no. 2, pp. 56–67, 2019. doi: 10.22437/pena.v9i2.7641.

[20] S. Rumilah and I. Cahyani, ”Struktur bahasa: pembentukan kata dan morfem sebagai proses morfemis dan morfofonemik dalam bahasa indonesia,” *Jurnal Pendidikan Bahasa Indonesia*, vol. 8, no. 1, pp. 70–87, 2020.

[21] Napitupulu, “UNIVERSITAS SUMATERA UTARA Poliklinik UNIVERSITAS SUMATERA UTARA,” J. Pembang. Wil. Kota, vol. 1, no. 3, pp. 82–91, 2017

[22] D. Novitasari, ”Perbandingan algoritma stemming porter dengan arifin setiono untuk menentukan tingkat ketepatan kata dasar,” *STRING (Satuan Tulisan Riset dan Inovasi Teknologi*, vol. 1, no. 2, pp. 120–129, 2016. doi: 10.30998/string.v1i2.1031.

[23] N. W. Wardani and P. G. S. C. Nugraha, ”Stemming dokumen teks bahasa bali Dengan metode rule base approach,” *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 7, no. 3, pp. 510–521, 2020. doi: 10.35957/jatisi.v7i3.538.

[24] M. Fikry, ”Stemmer bahasa melayu riau berdasarkan aturan morfologi,” in *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI) 13*, Nov. 18, Pekanbaru, Indonesia, 2021, pp. 118–124.