Recursive Journal of Informatics
Vol. 2, No. 2, September 2024

p-ISSN 2963-5551 https://journal.unnes.ac.id/journals/index.php/rji e-ISSN 2986-6588

Implementation of Raita Algorithm in Manado-Indonesia Translation Application with Text Suggestion Using Levenshtein Distance Algorithm

Novanka Agnes Sekartaji[1], **Riza Arifudin**[2]
[1,2]Informatics Engineering Study Program, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia

Abstract. Manado City is one of the multidimensional and multicultural cities, possessing assets that are considered highly potential for development into tourism and development attractions. The current tourism assets being developed by the Manado City government are cultural tourism, as they hold a charm and allure for tourists. Hence, a communication tool in the form of a translation application is necessary for facilitating communication between visiting tourists and the native community of North Sulawesi, even for newcomers who intend to reside in North Sulawesi, given that the Manado language serves as the primary communication tool within the community. This research employs a combination of the Raita algorithm and the Levenshtein distance algorithm in its creation process, along with the confusion matrix method to calculate the accuracy of translation results using the Levenshtein distance algorithm with a text suggestion feature. The research begins by collecting a dataset consisting of Manado language vocabulary and their translations in Indonesia language, sourced from literature studies and original respondents from North Sulawesi, which have been validated by a validator to prevent translation data errors. The subsequent stage involves preprocessing the dataset, converting the entire content of the dataset to lowercase using the case folding process, and removing spaces at the start and end of texts using the trim function. Next, both algorithms are implemented, with the Raita algorithm serving for translation and the Levenshtein distance algorithm providing text suggestions for typing errors during the translation process. The accuracy results derived from the confusion matrix calculations during the translation process of 100 vocabulary words, accounting for typing errors, indicate that the Levenshtein distance algorithm is capable of effectively translating vocabulary accurately and correctly, even in the presence of typing errors, resulting in a high accuracy rate of 94,17%.
Purpose: To determine the implementation of the Levenshtein distance and Raita algorithms in the process of using the Manado-Indonesian translation application, as well as the resulting accuracy level.
Methods/Study design/approach: In this study, a combination of the Raita and Levenshtein distance algorithms is utilized in the translation application system, along with the confusion matrix method to calculate accuracy.
Result/Findings: The accuracy achieved in the translation process using text suggestions from the Levenshtein distance algorithm is 94.17%.
Novelty/Originality/Value: This research demonstrates that the combination of the Raita and Levenshtein distance algorithms yields optimal results in the vocabulary translation process and provides accurate outcomes from the use of effective text suggestions. This is attributed to the fact that nearly all the data used was successfully translated by the system, even in the presence of typographical errors.
Keywords: Manado City, Translation Application, Manado-Indonesia Language, Raita Algorithm, Levenshtein Distance Algorithm, Text Suggestion, Confusion Matrix.
Received August 22, 2023 / **Revised** May 02, 2024 / **Accepted** September 17, 2024

INTRODUCTION

According to legend, the origin of the city of Manado comes from "Wanua Wenang" which means that this person is a native of Minahasa [1]. The word Manado itself is the name of an island located opposite the island of Bunaken, which in Minahasa is "Mana rou" or "Mana dou" which means "far away [2]. Being a multicultural and multidimensional city, Manado possesses an advantage in terms of assets that are evaluated as highly potential for development into tourism attractions, aiming to enhance the local revenue. The current focus of the Manado City government's tourism asset development is cultural tourism, driven by the captivating allure it holds for both local and international tourists [3]. Most of the

[1]*Novanka Agnes Sekartaji
Email addresses: novankasekartaji@students.unnes.ac.id (Sekartaji)
DOI: 10.15294/rji.v2i2.73651

Manado language is the same as the vocabulary in Indonesian in general, but there are some fundamental differences, namely the loan words from Dutch and Portuguese which are absorbed into the use of the Manado language, as well as the use of the vocabulary "we" which functions as a the first person singular pronoun or in Indonesian is usually called me or me [4]. As a result, the development of cultural tourism underscores the importance of the role of the local language in facilitating communication between the native population of North Sulawesi, particularly those engaged as traders working within tourist destinations, and local as well as foreign tourists who are interested in learning the local language, Manado language.

The increasing prevalence of the use of the Manado language in daily life compels newcomers and prospective tourists visiting Manado as a tourist city to adapt to communicating in Manado language. This adaptation is crucial to avoid being perceived as outsiders and to ensure seamless communication during their exploration of Manado. The utilization of Manado language by its users, harmoniously coexisting with Indonesian, regional languages, and foreign languages, signifies that Manado language serves as the primary means of communication within the community [5].

The existence of an electronic dictionary can serve as a means of communication between tourists and the native community as well as newcomers in Sulawesi Utara. The development of this application will employ two algorithms: the Raita algorithm, which functions to translate vocabulary, and the Levenshtein distance algorithm, which provides text suggestions or word recommendations based on typing errors made by users of the application.

The raita algorithm is a part of the string matching algorithm which in its implementation can be used to compare pattern characters with text characters which in the matching process start from the rightmost character [6]. Raita created an algorithm that implements it by comparing the last character of the pattern to the character on the right side of the window until a match occurs. Raita observes that this algorithm has good results in practice when searching for patterns in English text and associates this search process with the dependence of pattern characters [7].

The Levenshtein distance algorithm belongs to the approximate string matching category, where in its function is to perform matching processes for specific word patterns within a sentence. This matching process relies on comparing the similarity between the writing of two strings to determine the degree of difference [8]. Another understanding explains that the Levenshtein distance algorithm is an algorithm which in use has a function to measure the value of similarity or compatibility between two words or strings [9].

METHODS

The methodology employed in this research involves several stages. It begins with a preprocessing phase, which includes case folding and trimming. Subsequently, the implementation of the Raita and Levenshtein distance algorithms takes place, alongside the utilization of the confusion matrix method during the accuracy calculation of the generated text suggestions.

1. Preprocessing Step

Preprocessing of the data is necessary to ensure that the dataset utilized for the study does not contain errors, missing values, inconsistent data, or unbalanced data.

a. Case folding

Case folding is one of the preprocessing stages that functions to convert all uppercase letters in the data into lowercase [10]. In the case folding stage, there is a process that transforms all vocabulary within the database and that will be displayed in the application into lowercase. For instance, the term "Kucing" becomes "kucing", uppercase characters are changed to lowercase letters during the case folding process [11].

b. Trim

The second preprocessing stage performed in this application is trimming. The trimming preprocessing stage serves to remove leading and trailing spaces from the vocabulary resulting from the application's translation process [12]. An example of the use of trim is with the vocabulary "   makan   ," which becomes "makan" after trimming removes the spaces preceding and following the word, thanks to the trim function.

2. Implementation of the Levenshtein Distance Algorithm
In the process of applying the Levenshtein distance algorithm, there is a step that generates text suggestions or word recommendations based on typing errors made by users of this application. The outcome of the Levenshtein distance algorithm calculation is obtained through a matrix computation, which is used to calculate the count of differences between two strings [13]. The computation operations of the Levenshtein distance algorithm encompass insertion, deletion, and substitution. The auto-correction feature also uses the Levenshtein distance algorithm, this is often seen in search engines on the internet. For example, when an internet user wants to type a word like "journal", while typing is still "journal", the search engine will suggest several words like "journal", "journal", or "journalist" as the word to search for [14].

3. Implementation of the Raita Algorithm
In the process of implementing the Raita algorithm, there is a step involving the comparison of characters located at the last position of the pattern to characters situated at the far-right position within the window. Subsequently, there is a process of comparing the pattern from the far-right position to the left until a match occurs. When a match is found, the character at the first position of the leftmost text pattern from the window is also compared. If a match occurs, the final step involves comparing the character at the middle position of the pattern to the text character located in the middle of the window [15].

4. Implementation of the Confusion Matrix
Confusion matrix is one of the methods used to perform good level analysis [16]. The function of the confusion matrix in this study is to calculate the accuracy value of 100 sample data vocabulary using the text suggestion feature from the Levenshtein distance algorithm during the translation process for vocabulary containing typing errors. Accuracy is the ratio of correct predictions to the entire data, precision is a comparison of the value of the correct positive predictions compared to the total results with positive predictions [17]. The formula for calculating accuracy using this method can be seen in Equation 1.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + True\ Positive + True\ Negative} \quad (1)$$

RESULT AND DISCUSSION
This research was conducted to develop a vocabulary translation application that incorporates a text suggestion feature during the translation process. The inclusion of this text suggestion feature enables the application to provide word recommendations for typing errors made by users. The research commenced with the processing of collected dataset, consisting of Manado language vocabulary and their translations in Bahasa Indonesia. This data collection period spanned one month, from November 7th 2022 to December 7th 2022. For this study, the dataset was sourced from websites and respondents originating from Sulawesi Utara. By December 7th 2022, the accumulated dataset comprised 851 Manado language vocabulary entries and their corresponding 903 translations in Bahasa Indonesia. Subsequently, this data was stored in CSV format and imported into MySQL. A database generation application called MySQL is open source, allowing anybody to use, develop, and use it on any operating system, including Windows, Linux, and Mac OS [18].

The next step involved importing this data into the application, which underwent two preprocessing stages: case folding and trim. An example of using case folding in this application is when a user inputs the vocabulary "Makan." In this case, the application's output will be "makan," as shown in Figure 1.

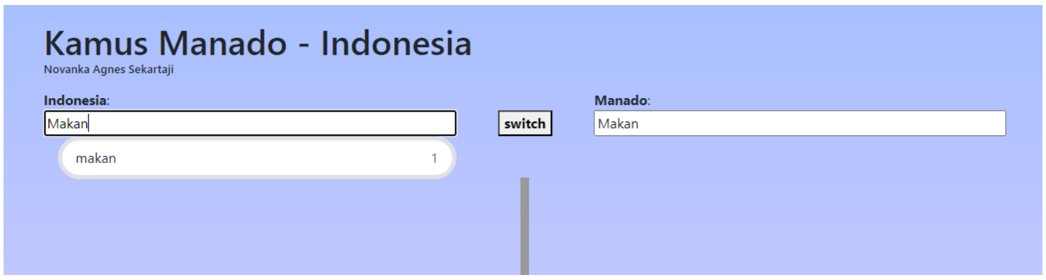


Figure 1. Case Folding Preprocessing

And an example of using trim on the vocabulary " makan " results in "makan," as depicted in Figure 2.

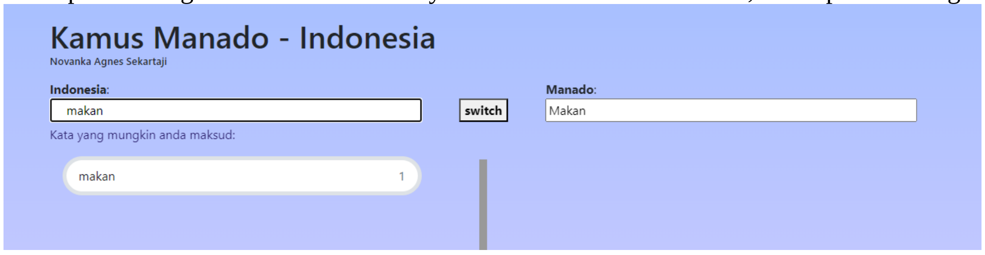

Kamus Manado - Indonesia
Novanka Agnes Sekartaji
Indonesia: makan
switch
Manado: Makan
Kata yang mungkin anda maksud:
makan 1

Figure 2. Trim Preprocessing

The outcome after undergoing the trim preprocessing stage can be observed in Figure 3.

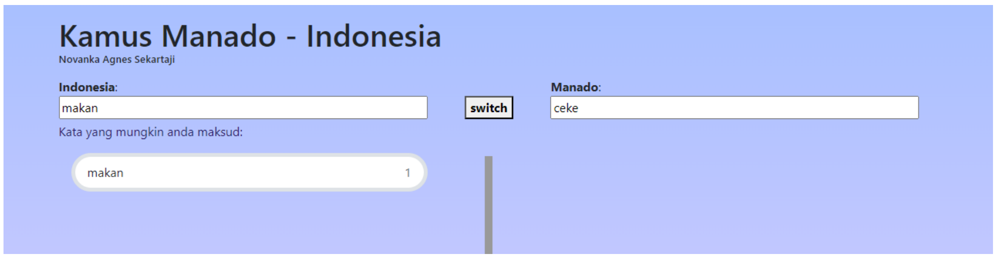

Kamus Manado - Indonesia
Novanka Agnes Sekartaji
Indonesia: makan
switch
Manado: ceke
Kata yang mungkin anda maksud:
makan 1

Figure 3. Trim Preprocessing Result

After completing the subsequent preprocessing stage, the first translation process will be conducted by the Raita algorithm. In the process of implementing the Raita algorithm, there is a step that involves comparing characters located at the last position of the pattern with characters situated at the far-right position within the window. Subsequently, there is a process of comparing the pattern from the far-right position to the left until a match occurs. When a match is found, the character at the first position of the leftmost text pattern from the window is also compared. If a match occurs, the final step involves comparing the character at the middle position of the pattern with the text character located in the middle of the window. Here's an example of the calculation using the Raita algorithm for Manado-Indonesian translation on the string: "ijo.".

Text (T)	= MARIJO
Pattern (m)	= IJO

The next step involves the creation of the BmBc table using Equations 2 and Equations 3 :

$$m - 2 = \ldots\ldots\ldots\ldots\ldots\ldots\ldots \tag{2}$$

Serving as the boundary for character search within the pattern.

$$m - i - 1 = \ldots\ldots\ldots\ldots\ldots\ldots\ldots \tag{3}$$

Serving to find the character value in the BmBc Table 1.

Table 1. Index Pattern

index (i)	0	1	2
pattern (P)	I	J	O

Based on Table 1, the calculation of the table can be determined using Equation (1):

$$m - 2 =$$
$$3\text{-}2 = 1$$
$$So,\ i = 0\text{-}1$$

Next, to find the shift value in the BmBc table, the formula in Equation (2) is used:

$$m - i - 1 =$$
$$\text{BM[BC[I]]} = 3 - 0 - 1 = 2$$
$$\text{BM[BC[J]]} = 3 - 1 - 1 = 1$$

(*) unrecognized character.

The value O corresponds to the length of the pattern, which is 3. Since the alphabet is not found in the table, it will be initialized with the symbol (*) and then assigned a value equal to the length of the pattern. Therefore, for the calculation of the Raita Algorithm according to the BmBc Table 2.

Table 2. BmBc Result (Pattern)

index	BC	BM
0	I	2
1	J	1
2	*	3

After the process that generates the BmBc table in Table 2, the next step is to perform the process of matching pattern characters against text characters.

1. First Iteration
This involves matching the last character of the pattern against the aligned last character of the text. If a mismatch occurs, the pattern's characters will shift to the right by the value indicated in the BmBc Table 3.

Table 3. First Iteration

Text	M	A	R	I	J	O
Pattern	I	J	O			

In the process shown in Table 3, a mismatch occurs between the pattern character "R" and the character "O" in the text. Therefore, the pattern characters will be shifted to the right by the value indicated in the BmBc table (*), which is 3 steps.

2. Second Iteration
The second step is to shift by 3 steps to the right (according to the BmBc value for the character 'O'. If a mismatch occurs, the pattern characters will shift to the right by the value indicated in the BmBc Table 4.

Table 4. Second Iteration

Text	M	A	R	I	J	O
Pattern				I	J	O

As seen in the matching process in Table 4, a successful match between all characters in the pattern and the text is achieved. The entire character matching process using the Raita algorithm has concluded and completed in the second iteration. This outcome corresponds to the result in the system, as depicted in Figure 4.

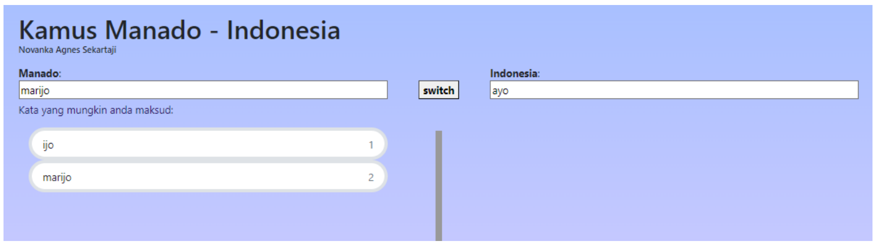

Kamus Manado - Indonesia
Novanka Agnes Sekartaji
Manado: marijo
switch
Indonesia: ayo
Kata yang mungkin anda maksud:
ijo 1
marijo 2

Figure 4. Translation Results in the Application Found in the 2nd Iteration

The subsequent step to address translation with typing errors will be managed by the Levenshtein distance algorithm. In the process of implementing the Levenshtein distance algorithm, there is a step that generates text suggestions or word recommendations for typing errors made by users of this application. The outcome of the Levenshtein distance algorithm calculation is obtained through a matrix computation, which is used to calculate the count of differences between two strings. To gain a better understanding of the manual calculation and the system's results, please refer to the example provided below.

Here is an example calculation of the Levenshtein distance algorithm for Manado-Indonesian translation on the string "pvor" using the formula from the Equation. We can calculate the Levenshtein distance for strings A and B using the following matrix:

a = 1st string => pvor $\quad$ i = 1
b = 2nd string =>ovor $\quad$ j = 1

The procedure for calculating Levenshtein distance is performed by populating the entire matrix, starting from the top-left corner as shown in Table 5.

Table 5. Character Sequence Initiation

		o	v	o	r
	0	1	2	3	4
p	1				
v	2				
o	3				
r	4				

The Levenshtein distance value is then placed in the bottom-right corner. Here is an example calculation: Here is an example calculation for the Levenshtein distance at position (1,1) in the matrix:

$$Lev_{a,b}(1,1) = min\begin{cases} Lev_{a,b}(0,1) + 1 = 1 + 1 = 2 \\ Lev_{a,b}(1,0) + 1 = 1 + 1 = 2 \\ Lev_{a,b}(0,0) + \ 1_{(if\ a_1 \neq b_1)} = 0 + 1 = 1 \end{cases}$$

$$Lev_{a,b}(1,1) = \begin{cases} 2 \\ 2 = 1 \\ 1 \end{cases}$$

Table 6 is an example calculation for the Levenshtein distance at position (1,1), where this distance is determined to be 1 based on the matrix calculation that has been performed.

Table 6. Lev Distance (1,1)

		o	v	o	r
	0	1	2	3	4
p	1	1			
v	2				
o	3				
r	4				

The next step is to calculate the Levenshtein distance at position (1,2), as shown in Table 7. Here's an example calculation for the Levenshtein distance with the values i = 1 and j = 2.

$$Lev_{a,b}(1,2) = min\begin{cases} Lev_{a,b}(0,2) + 1 = 2 + 1 = 3 \\ Lev_{a,b}(1,1) + 1 = 1 + 1 = 2 \\ Lev_{a,b}(0,1) + \ 1_{(if\ a_1 \neq b_1)} = 1 + 1 = 2 \end{cases}$$

$$Lev_{a,b}(1,2) = \begin{cases} 3 \\ 2 \ = 2 \\ 2 \end{cases}$$

Table 7 represents the calculation for the Levenshtein distance at position (1,2), which results in a Levenshtein distance of 2 for this calculation.

Table 7. Lev Distance (1,2)

		o	v	o	r
	0	1	2	3	4
p	1	1	2		
v	2				
o	3				
r	4				

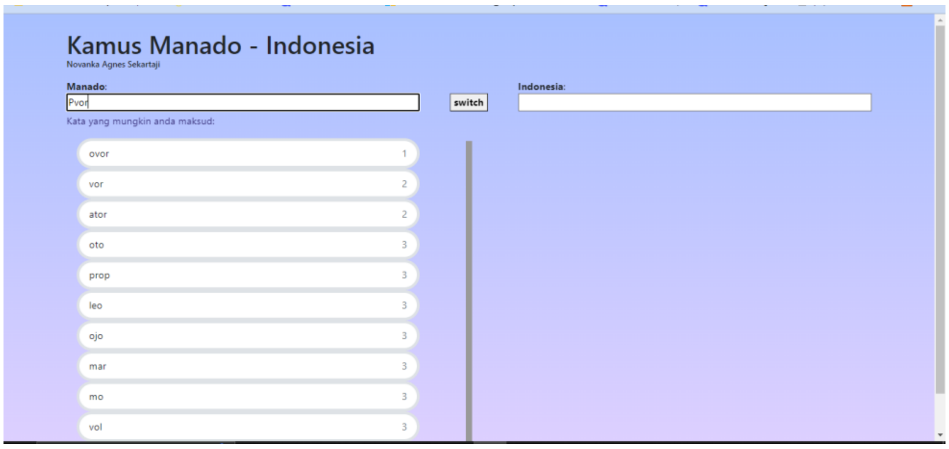
Next, the complete calculation of the Levenshtein distance will be performed up to position (4,4), as shown in Table 8. The final Levenshtein distance value is located at the bottom-right corner of the table.

Table 8. Lev Distance(4,4)

		o	v	o	r
	0	1	2	3	4
p	1	1	2	3	4
v	2	2	1	2	3
o	3	2	2	1	3
r	4	3	3	2	1

Lev Distance: 2

Table 8 represents the result of the Levenshtein distance calculation from the above example between the first string "pvor" and the second string "ovor" is the number located at the bottom-right corner, which is 1. In the above process, there are two operations in the Levenshtein distance algorithm: insertion operation for the character 'o' in the string 'pvor' and deletion operation for the character 'p' in the string 'pvor'. After these two operations are performed, the string 'pvor' transforms into 'ovor', which in Manado language means 'oper'. For the system's result regarding the text suggestion or word recommendation for the typing error in the vocabulary "pvor" becoming "ovor," you can refer to Figure 5.



Kamus Manado - Indonesia

Novanka Agnes Sekartaji

Manado: Pvor

Indonesia:

Kata yang mungkin anda maksud:

ovor	1
vor	2
ator	2
oto	3
prop	3
leo	3
ojo	3
mar	3
mo	3
vol	3

Figure 5. Levenshtein Distance Results on Application

After completing the translation process, the next step is to calculate the accuracy of the suggested words during the translation of vocabulary with typing errors using the Levenshtein distance algorithm. The translation results for 100 Manado vocabulary words to Indonesian, with typing errors, produce several values: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These values will be used to evaluate the performance of the Levenshtein distance algorithm in providing word suggestions for typing errors and to calculate various evaluation metrics, such as accuracy. This is shown in the Table 9.

Table 9. TP, TN, FP, dan FN Values

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP: 97	FP: 3
	Negative	FN: 3	TN: 0

From the table, the accuracy value is then calculated as follows.

$$\text{Accuracy} = \frac{(100 + 0)}{(97 + 0 + 3 + 3)} \times 100\%$$

$$\text{Accuracy} = 94{,}17\%$$

From the obtained values of TP, FN, FP, and FN in the confusion matrix calculation for the translation of 100 vocabulary words, the resulting accuracy can be seen in Table 9. This table demonstrates that by using the Levenshtein distance algorithm, the application is capable of translating vocabulary accurately even in the presence of typing errors, resulting in a high accuracy rate of 94.17%.

CONCLUSION

Based on the research findings and discussion regarding the study on the implementation of the Raita algorithm in a translation application for Manado-Indonesian language with text suggestion using the Levenshtein distance algorithm, the following conclusions can be drawn.

1. The application of the Raita algorithm is used in the translation process, while the Levenshtein distance algorithm is utilized in the text suggestion feature to translate vocabulary with typing errors made by users of the application. After the translation process, the confusion matrix method is employed to calculate the accuracy of the translation process for vocabulary with typing errors using the Levenshtein distance algorithm. The combination of these methods aims to enhance the effectiveness of vocabulary lookup.
2. This research also aimed to calculate the accuracy of translation with text suggestion using the Levenshtein distance algorithm. The accuracy achieved in this study, based on a sample of 100 Manado vocabulary words translated to Indonesian with typing errors, using the confusion matrix calculation on the text suggestion results of the Levenshtein distance algorithm is 94,17%.

Based on the results of the research that has been described regarding the creation of this translation application, the advice given to be able to carry out even better development is that in the translation process with the Raita algorithm there was a system error in some vocabulary due to continuous iteration processes, for example vocabulary with character length the pattern is longer than the text, for example when the user types "marijono" with the meaning of the word "ijo", the system will error and the second is that the number of datasets that belong to a translation application is small because it only consists of less than 1000 vocabularies, resulting in overfitting or a condition where the algorithm that works on the training data memorized system but is unable to generalize to new data so that it often results in low accuracy results. This can be done by increasing the number of datasets which can be done by increasing the number of literature studies directly at the North Sulawesi provincial language center, so that more data is collected and there is no need to process data and re-validate which takes quite a long time, up to months.

REFERENCES

[1] V. V. Mengko, N. Kandowangko, and L. Lesawengen, “Kehidupan waria di kota manado,” *J. Acta Diurna Komun.*, vol. 5, no. 4, p. 12, 2016, [Online]. Available: ttps://ejournal.unsrat.ac.id/v3/index.php/actadiurnakomunikasi/article/view/13208

[2] M. R. K. Damopoli, “Aset Tanah dan Bangunan Pemerintah Daerah Kota Manado,” *J. Ilmu Polit.*, vol. Vol. 9, no. 2, pp. 1–11, 2020, [Online]. Available: https://ejournal.unsrat.ac.id/index.php/politico/article/view/30710

[3] F. Kerebungu, “Pengembangan industri pariwisata budaya dalam meningkatkan pendapatan asli daerah (PAD) kota Manado,” *Jurnal Aplikasi Manajemen*. pp. 289–295, 2020. [Online]. Available: https://www.jurnaljam.ub.ac.id/index.php/jam/article/view/1880

[4] E. Lapian, A. B. Osmond, and R. E. Saputra, “Recurrent neural network untuk pengenalan ucapan pada dialek Manado,” in *e-Proceeding of Engineering*, 2018, vol. 5, no. 3, pp. 6436–6443. [Online]. Available: https://core.ac.uk/display/299925133

[5] V. (Universitas S. R. Kamu and I. J. (Universitas S. R. Moniung, “Morfologi bahasa Melayu Manado,” *J. Kaji. Linguist.*, vol. 9, no. 2, pp. 248–253, 2021, doi: 10.35796/kaling.9.2.2021.38945.

[6] N. Marbun, M. Zarlis, D. Hartama, Mesran, and B. J. . Sitompul, “Implementasi algoritma raita pada pencarian katalog alkes,” in *Seminar Nasional Sains & Teknologi Informasi (SENSASI)*, 2019, pp. 520–523. [Online]. Available: http://prosiding.seminar-id.com/index.php/sensasi/article/view/357

[7] C. Charras and T. Lecroq, *Handbook of exact string matching algorithms*. College Publications, 2004. [Online]. Available: https://www.researchgate.net/publication/220693416 Handbook

[8] V. Sahfitri and I. B. Zarizal, “Approximate string matching untuk pencarian kata dalam kamus bahasa indonesia,” *J. Ilm. MATRIK*, vol. 24, no. 3, pp. 248–259, 2022.

[9] M. J. Tannga, S. Rahman, and Hasniati, “Analisis perbandingan algoritma levenshtein distance dan jaro winkler untuk aplikasi deteksi plagiarisme dokumen teks,” *J. Technol. Res. Inf. Syst. Eng.*, vol. 4, no. 2, pp. 44–54, 2017, [Online]. Available: https://jurnal.kharisma.ac.id/jtriste/article/view/29

[10] F. Pramono, Didi Rosiyadi, and Windu Gata, “Integrasi n-gram, information gain, particle swarm optimation di naïve bayes untuk optimasi sentimen Google Classroom,” *J. Rekayasa Sist. dan*

Teknol. Informasi) (RESTI), vol. 3, no. 3, pp. 383–388, 2019, doi: 10.29207/resti.v3i3.1119.

[11] K. Munawaroh and A. Alamsyah, “Performance comparison of SVM, naïve bayes, and KNN algorithms for analysis of public opinion sentiment against COVID-19 vaccination on Twitter,” *J. Adv. Inf. Syst. Technol.*, vol. 4, no. 2, pp. 113–125, 2023, doi: 10.15294/jaist.v4i2.59493.

[12] A. Saputra and F. Noor Hasan, “Analisis sentimen terhadap aplikasi coffee meets bagel dengan algoritma naïve bayes classifier,” *SIBATIK J. J. Ilm. Bid. Sos. Ekon. Budaya, Teknol. dan Pendidik.*, vol. 2, no. 2, pp. 465–474, Jan. 2023, doi: 10.54443/sibatik.v2i2.579.

[13] M. M. Yulianto, R. Arifudin, and A. Alamsyah, “Autocomplete and spell checking levenshtein distance algorithm to getting text suggest error data searching in library,” *Sci. J. Informatics*, vol. 5, no. 1, p. 75, 2018, doi: 10.15294/sji.v5i1.14148.

[14] A. P. U. Siahaan *et al.*, “Combination of levenshtein distance and rabin-karp to improve the accuracy of document equivalence level,” *Int. J. Eng. Technol.*, vol. 7, no. 2 Special Issue 27, pp. 17–21, 2018, doi: 10.14419/ijet.v7i2.27.12084.

[15] K. Al-khamaiseh and S. AlShagarin, “A survey of string matching algorithms,” *Int. J. Eng. Res. Appl.*, vol. 4, no. 7, pp. 144–156, 2014, [Online]. Available: https://www.researchgate.net/publication/279442892

[16] L. Ariyanti and A. Alamsyah, “C4.5 Algorithm Optimization and Support Vector Machine by Applying Particle Swarm Optimization for Chronic Kidney Disease Diagnosis,” *Recursive J. Informatics*, vol. 1, no. 1, pp. 18–26, 2023, doi: 10.15294/rji.v1i1.65196.

[17] A. Ridhovan and A. Suharso, “Penerapan metode residual network (Resnet) dalam klasifikasi penyakit pada daun gandum,” *J. Ilm. Penelit. dan Pembelajaran Inform. (JIPI)*, vol. 7, no. 1, pp. 58–65, 2022, doi: 10.29100/jipi.v7i1.2410.

[18] N. Wakhidah and M. Mudi, “The application of simple additive weighting method in the selection of the islamic competition winners,” *J. Adv. Inf. …*, vol. 1, no. October, pp. 1–12, 2019, [Online]. Available: https://journal.unnes.ac.id/sju/index.php/jaist/article/view/36497%0Ahttps://journal.unnes.ac.id/sju/index.php/jaist/article/download/36497/15063