Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13856
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Transliteration of Latin Letters to Bali Characters Based on Unicode for Mobile Devices using Finite State Automata and Levenshtein Distance

I Made Subrata Sandhiyasa[1)*] , Ni Kadek Nita Noviani Pande[2)] , Luh Fibriyanthini [3)]
[1,2)]Institut Bisnis dan Teknologi Indonesia, Denpasar, Indonesia
[3)] SMK Negeri 1 Denpasar, Indonesia
[1)*]subrata.sandhiyasa@instiki.ac.id ,[2)] novy.pande@instiki.ac.id , [3)]fibriyanthini@gmail.com

Submitted : Jul 11, 2024 | **Accepted** : Aug 1, 2024 | **Published** : Aug 4, 2024

Abstract: The preservation of Balinese script writing has been pursued by the local government with the issuance of the Bali Province Regional Regulation Number 1 of 2018 concerning Balinese Language, Script, and Literature. However, the use of Balinese script in daily life is declining, especially among generation Z, most of whom find Balinese language difficult to use and have never been taught it. Technological advances, particularly smartphone technology, can play an important role in shaping generation Z's habits, values, and social interaction patterns. This research uses the Finite State Automata (FSA) method to convert Latin letters to the Balinese script Unicode standard, following the Balinese script writing rules. FSA governs transliteration behavior by using the working principles of State, Event, and Action. Besides transliterating sentences typed by users, the application produced by this research also displays Balinese script words related to the words typed by users using the Levenshtein Distance method. The 'related words' feature allows users to know more about Balinese script than just the typed word. From the test results conducted through two different test cases, the first test case tested the application's ability to transliterate words/sentences typed by users without selecting words from the application's suggestions. The results showed that of the 50 words tested, 39 were correctly transliterated. The second test case tested the app's ability when the user selects a word from the suggestions given by the app. The result shows that out of 50 words tested, 43 transliteration data are correct, with the total accuracy of both test cases being 82%.

Keywords: Bali Characters; Generation Z; Mobile Technology; Finite State Automata; Levenshtein Distance

INTRODUCTION

Balinese script consists of Wréastra, Swalalita, and Modré. The Wrésastra script, which is more popularly known as anacaraka in Bali, has 18 pieces and the Swalalita script is the script used in Kawi literature, such as in writing jejawian and parwa. The number of Swalalita Script is 47 and is divided into 14 vowels and 33 consonants. Modré script is a sacred script used in mantras (Birawidya et al., 2022).

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13856 p-ISSN : 2541-044X

The preservation of Balinese writing has been carried out by the local government with the issuance of the Regional Regulation of Bali Province Number 1 of 2018 concerning Balinese Language, Script and Literature (Widiantana, 2022). However, the problem of the use of Balinese script in everyday life is declining with the times, a survey conducted on Gen-Z showed that 87% considered Balinese difficult to use while 13% of them admitted that they were never taught Balinese. Similarly, in daily life only 27% of children use Balinese and 73% use Indonesian (Utari & Ristiani, 2016).

On the other hand, the advancement of technology, especially smartphone technology, is not only a means of communication, but can play an important role in shaping the habits, values, and social interaction patterns of the Z generation. This can be utilised as a momentum to preserve Balinese Script by utilising mobile technology combined with word processing methods that are currently developing. The urgency of this research is 1) The decreasing interest and knowledge of the younger generation, especially Generation Z in the use of Balinese language and script (Devi & Sudarma, 2023). 2) The lack of digital platforms intended to educate Balinese characters in line with the interests of the younger generation who are accustomed to using smartphones in their daily lives.

Mobile technology, which has become an important part of generation Z's life, can be utilised to support the preservation of Balinese script. This research aims to develop an Android-based mobile application that is able to transliterate Latin letters into Balinese script using the Unicode standard (Standard, 2023). The method used is Finite State Automata (FSA), which has been proven effective in transliteration algorithms, even for complex writing rules (Astuti et al., 2022). FSA governs transliteration behaviour by using the working principles of state, event, and action. In addition, the app is also equipped with a 'related words' feature that uses the Levenshtein Distance method to display Balinese script words related to the word typed by the user. This Levenshtein Distance method has been proven to have a high level of accuracy in finding related words (Azhri et al., 2019; Rosmala & Risyad, 2018a).

This research aims to build a transliteration application of Latin letters to Balinese script based on mobile android using unicode as a character coding standard. For transliteration accuracy, this research uses the Finite State Automata method which has been proven effective in transliteration algorithms, even in complex rules (Prabhakar & Pal, 2018). As a means to add insight to the user, this research will display related words that are identical to what is typed by the user using the Levenshtein distance character similarity detection method. The contribution of the research is focused on providing a digital platform that can be accessed by the younger generation, making it easier for them to learn and use Balinese script through technology that is already familiar to them. Thus, this research not only seeks to preserve the Balinese script, but also integrates modern technology to support its sustainability amidst the advancement of the times.

LITERATURE REVIEW

Finite State Automata

Finite State Automata (FSA) is a machine that accepts a string of symbols from a particular alphabet as input and has a finite number of states. The initial state is the state where the machine is before processing the input, and the final state is the state where the machine is after processing the entire input. FSAs can be either Deterministic Finite Automata (DFA) or Non-deterministic Finite Automata (NFA) (Chowdhury et al., 2013; Gold & Zesch, 2020). An FSA has an infinite number of states and can move from one state to another. This state change is expressed by a transition function. FSAs have no storage, so their 'remembering' ability is limited, only remembering the current state. Examples of FSAs include elevators, text editors, lexical analyzers, network communication protocols and parity checkers. FSAs based on defining their state-changing capabilities can be divided into Deterministic Finite Automata (DFA) and Non-deterministic Finite Automata (NFA).(Studi et al., 2011). Formally FSA is expressed by 5 tuples or M = (Q, ∑, δ, S, F) where:

Q = set of states / positions
∑ = set of input symbols/alphabets
δ = transition function
S = initial state, S ϵ Q
F = set of final states, F ∩ Q (the number of final states in an FSA can be more than one)

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13856 p-ISSN : 2541-044X

DFA works by reading the input string one symbol at a time and moving from one state to another according to a transition function. The DFA accepts a string if, after reading the entire input, the machine is in one of the final states. Figure 1 is a table describing the transition function of the FSA state automaton. The table shows, for example, if the FSA is in state p and consumes 1, then the FSA will move to state q. FSAs actually come in two forms, depending on the properties you need from the transition function. First of all, we will look at a class of FSAs called deterministic finite state automata (DFA). In these machines, the current state of the machine and the current input symbol together determine exactly the final state of the machine: for every pair <current state, current input symbol>, there is exactly one possible next state for the machine.

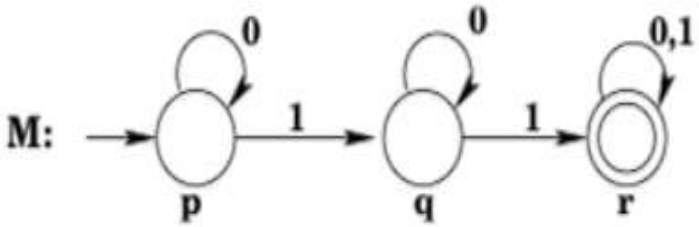

Fig. 1 DFA Transition Diagram

FSA has been widely used in language transliteration including (Rachman et al., 2020) conducted Text to Speech research on the madura language, the process of cutting words into syllables was carried out using the two-level Finite State Automata method. The results of the first level FSA become input for the second level FSA. FSA applications in text to speech applications are effectively used with an accuracy value of 90%. Furthermore (Belkhir et al., 2019; Karmakar et al., 2019) This study proposed a new machine translation method based on a stochastic finite-state model for spoken English. This method formalizes a rational grammar using stochastic finite-state. Through a given pair of source and target sentences, the method generates a set of conventional rules from which a stochastic rational grammar will be inferred, and the grammar is finally converted into a finite-state automaton. FSA is also used in various applications, including aspects of natural language processing (NLP)(Mahe et al., 2024) .

Lavenstein Distance

Using the Levinshtein string matrix, we can measure the difference or distance between two strings. The distance value between two strings is determined by the number of change operations required to transform a string from one string to another. Insertion, deletion, and swapping are operations. One of the Levinshtein distance algorithms can be used to find the similarity between two strings that may commit plagiarism(Aung, 2019).

The Lavenstein Distance algorithm is a weighting approach to assign a cost of 1 to each editing operation (Insertion, deletion and substitution). For example, the Levenshtein edit of the distance between "dog" and "cat" is 3 (replacing d with c, o with a, g with t) (Errattahi et al., 2018; Hládek et al., 2020). Levenshtein's algorithm helps determine the number of modifications, inserts and deletions on string c1 to be equal to string c2. It calculates the minimum number of basic operations executed. For this purpose, the algorithm uses an (n+1)*(m+1) dimensional matrix, where n and m are the lengths of the previous two strings.

There are three major operations that can be performed by this algorithm, namely character insertion operations to add certain characters to a string, then character substitution operations to replace certain characters in a string, and character deletion operations to remove certain characters in a string. To calculate the distance with Levenshtein distance can be done by using a table as shown in Figure 1.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13856 p-ISSN : 2541-044X

		k	i	t	t	e	n
	0	1	2	3	4	5	6
s	1	1	2	3	4	5	6
i	2	2	1	2	3	4	5
t	3	3	2	1	2	3	4
t	4	4	3	2	1	2	3
i	5	5	4	3	2	2	3
n	6	6	5	4	3	3	2
g	7	7	6	5	4	4	3

		S	a	t	u	r	d	a	y
	0	1	2	3	4	5	6	7	8
S	1	0	1	2	3	4	5	6	7
u	2	1	1	2	2	3	4	5	6
n	3	2	2	2	3	3	4	5	6
d	4	3	3	3	3	4	3	4	5
a	5	4	3	4	4	4	4	3	4
y	6	5	4	4	5	5	5	4	3

Fig 2. Levenshtein Distance Calculation Table (Chowdhury et al., 2013)

In Figure 1 there is a cell containing a number of letters checked. Table charging starts from cell (1,1) with row position as index i and column position as index j. Levenshtein-based table charging distance algorithm (Chowdhury et al., 2013).

METHOD

In the initial process there is a flow of research carried out, with the flow of stages intended to facilitate in explaining in general terms the process that has been carried out.

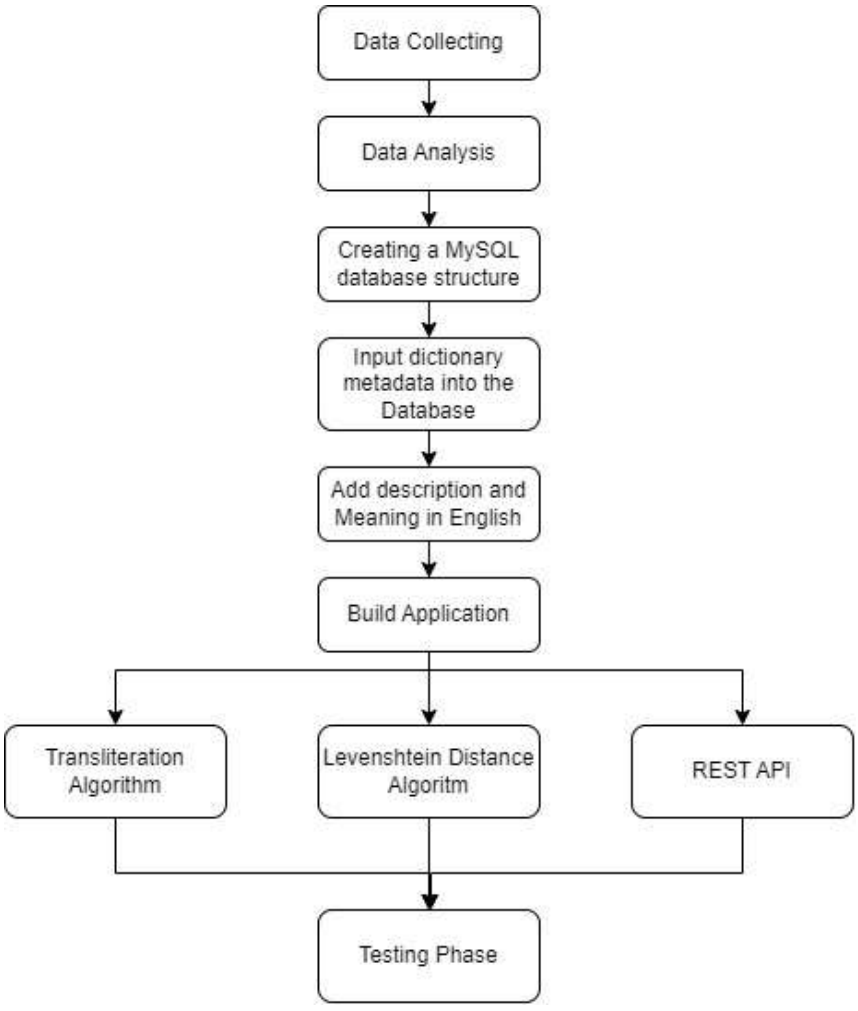

Fig 3. Research Stages

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13856 p-ISSN : 2541-044X

Based on Figure 3, it can be explained that the initial stage is to collect relevant data for transliteration, such as Latin-Bali dictionaries, Unicode characters, and transliteration rules. The data that has been collected is then analysed to understand the patterns and structures required for the transliteration of Latin letters to Balinese script. After the analysis, a MySQL database structure was created to store the dictionary data and transliteration metadata. The dictionary metadata, such as Latin and Balinese script pairs, were then entered into the created database. To provide a more complete context, the description and meaning of each transliteration pair was also added into the database. Once the data is ready, the next step is to build a mobile application that will be used to transliterate Latin letters to Balinese script based on the data that has been stored in the database. The implementation of a transliteration algorithm that uses Finite State Automata is done to convert Latin letters into Balinese script. In addition, the Levenshtein Distance Algorithm is used to handle errors or input variations that may occur during the transliteration process. To enable communication between the mobile application and the server that stores the database and processes transliteration requests, a REST API was developed. The last stage is accuracy testing.

Data Synchronization

Data synchronization is done the first time when the user installs the application as shown in Figure 3, this procedure uses the REST API method. Furthermore, the data will be stored in the smartphone's internal storage using the SQLite mobile database, so that when using the application, users do not depend on an internet connection. Users will only need the internet to use the application if there is a database update on the cloud server.

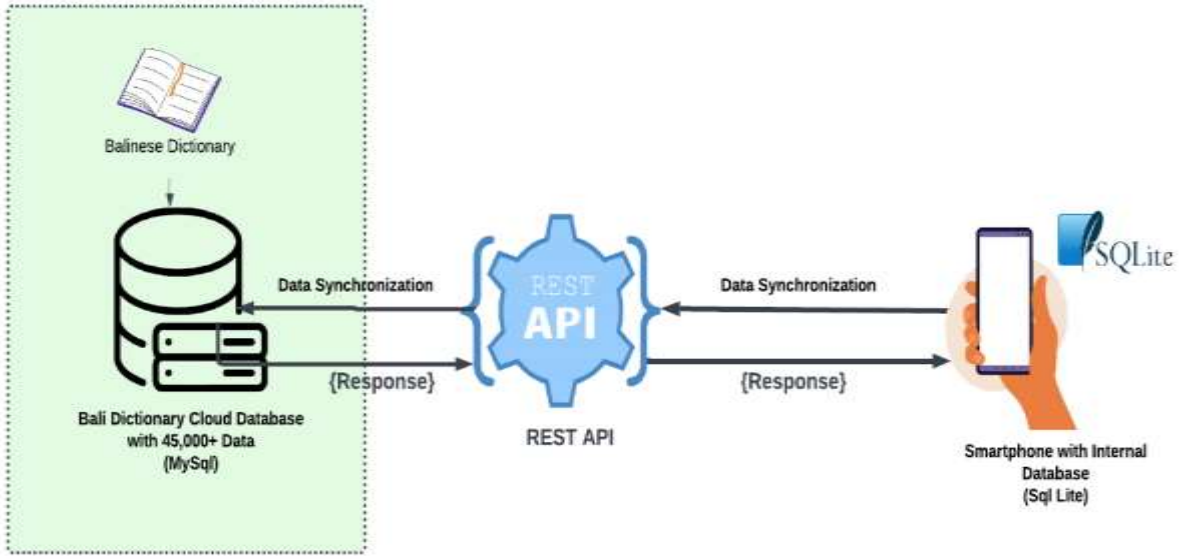

Fig. 4 Synchronize Dictionary
Source : author's document

Unicode-based Tranliteration of Latin Characters to Balinese Script

The Finite State Automata method in this study is used for transliteration and checking the series of letters in words/sentences in Latin letters, separating syllables from the series of letters and then the resulting syllables are converted into Balinese characters in Unicode format. So that later the unicode that has been assembled will be used to display Balinese characters with the Noto Sans Balinese font style that has been registered with Google Fonts. Iterations are performed using the Finite State Automata method as shown in Figure 4.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13856 p-ISSN : 2541-044X

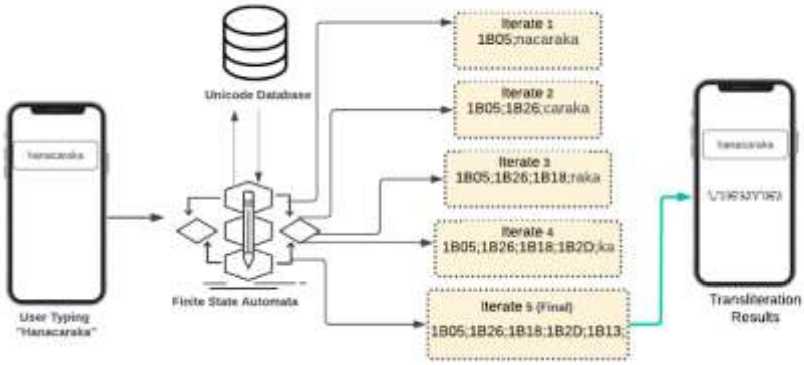

Fig. 5 Transliteration Process
Source : author's document

Seen in Figure 5 after the user inputs the Latin text "Hanacaraka", the application will run the Finite State Automata algorithm starting from the first state "Ha" and continuing until the last state of the word is "Ka". When switching states, if the state is in accordance with the rules set, then the FSA algorithm runs the function of replacing the Latin sentence using unicode. The function will continue to run until all the Latin letters in the sentence are converted into unicode and displayed on the user's screen.

Related word search using the levenshtein distance method

To find the relationship of words from user input with words contained in the smartphone's local database, this research uses the levenshtein distance method which has been tested to have a very high level of accuracy in terms of finding related words. The initial step of this method is done after the user types in a certain word, then the levenshtein distance method will measure the distance between strings and calculate the word similarity value in the sentence inputted by the user. This research applies a similarity value threshold of 50% or 0.5 and what will be displayed to the user is the 10 words that have the highest similarity. The workflow of the applied method is shown in Figure 6.

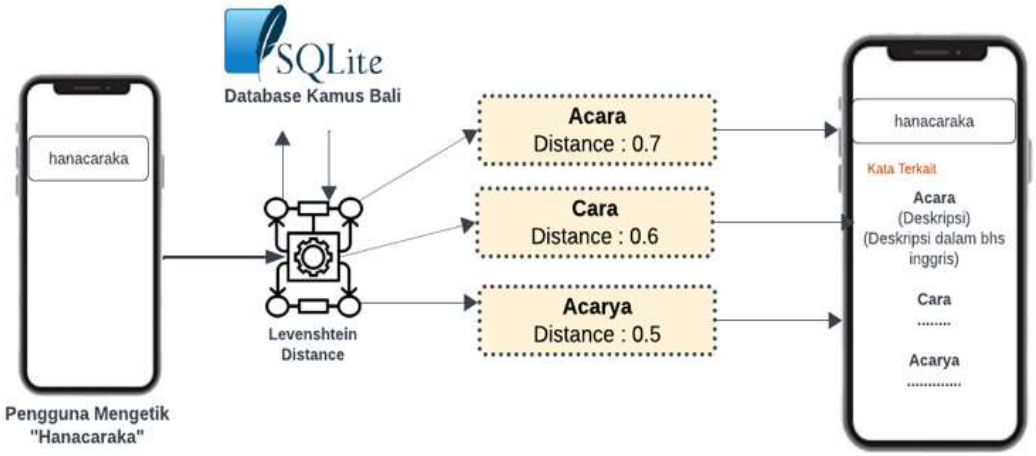

Fig. 6 Levenshtein distance method for related words search
Source : author's document

In Figure 6, it can be seen that when a user types a Latin word or selects a word in the smartphone's local database, the word will be compared to all other words in the local database. The work done by the levenshtein distance method in this study is a per-character replacement of each word being compared. As seen in Figure 5, the word "Hanacaraka" typed by the user is close to the word event with a distance of 0.7, cara with a distance of 0.6 and acarya with a distance of 0.5.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13856 p-ISSN : 2541-044X

RESULT

Unicode-based Tranliteration of Latin Characters to Balinese Script

In Figure 7, the transliteration application interface is shown, when the user types a word, the application will display a word suggestion that comes from the application's internal database after previously synchronizing with the data contained in the cloud database. the user can be seen typing the word "abejug" on the application's translate page. Then the application displays the selected word, the transliterated script of the word, the meaning in Indonesian and the meaning in English.

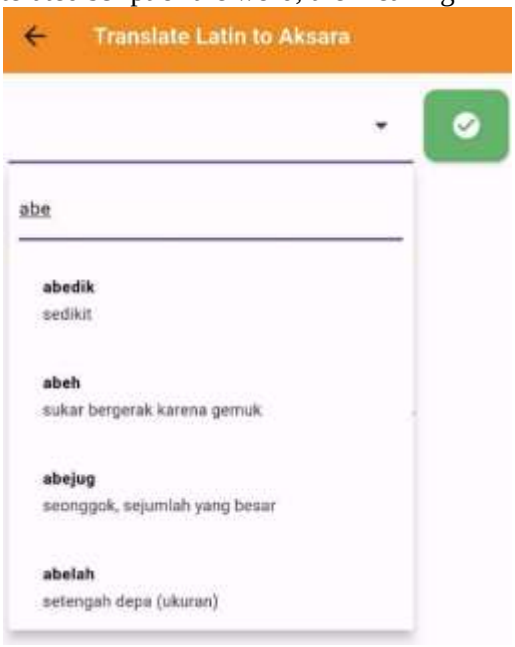

Fig. 7 Transliteration App Interface

Fig. 8 Transliteration of Latin words

The transliteration process performed on the word "abejug" using Finite State Automata is as follows,

1. Word Tokenization
   Checking the word fragments contained in the dictionary database. The characters of the word fragment "abejug". In this process the application performs tokenization, namely reading each syllable and space. This is done to characterize each word according to the characters contained in the database. The FSA process recognizes and captures patterns in the data. So that it can accept input and issue outputs that have an infinite number of states and can move from state to state.
   as shown in Figure 7 and the tokenization results are shown in Table 1.

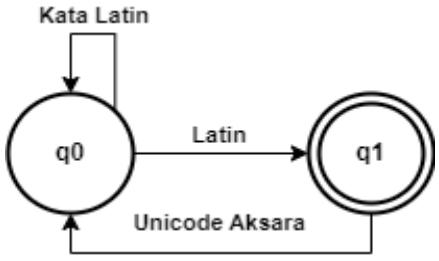

Fig. 9 Character separation in words

Where:
Q = set of states {q0, q1}
Σ = set of input symbols / input / alphabet {character characters}
S = initial state, S∈ Q = q0
F = final state set, F⊆ Q = q1

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13856 p-ISSN : 2541-044X

Table 1. Character Tokenization Results

Latin	Bali	Unicode
a	ha	1B33
b	ba	1B29
e	ae	1B42
j	ja	1B1A
u	u	1B38
g	ga	1B16

2. Conversion and customization of characters to unicode format
In this process, the conversion of Balinese script into a set of unicode of a Balinese script is separated by comma quotes. Furthermore, a complete set of unicode is adjusted based on the writing rules of the Balinese script. The unicode adjustment is done as much as the set of states contained in a word. This processing is as shown in Figure 8.

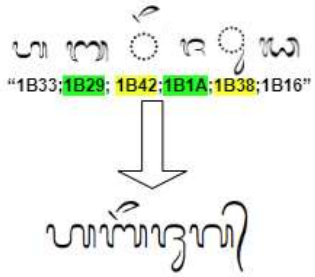

Fig.10 Character conversion and customization

3. Display results to the application interface
The last stage in the transliteration results is to display the results to the user interface. There are 2 types of results that will be displayed to the user, namely 1) results if the word typed by the user is found in the application dictionary database, then the results can be displayed in full with the meaning of the word in Indonesian and English. 2) results if the word typed by the user is not found in the application dictionary database, then the results displayed are only in the form of Latin transliteration to Balinese script as intended by Figure 9 and Figure 10.

Fig. 11 Displaying Full Results

Fig.12 Displaying Transliteration Only

Display related words

In addition to the results of Latin transliteration to Balinese script, meaning in Indonesian and meaning in English. In the results also displayed words related to what is typed by the user. This paper uses the Levenstein Distance method to compare adjacent words. The limit of the similarity value (threshold) is 50% or 0.5 and what will be displayed to the user is 10 words that have the

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.13856 p-ISSN : 2541-044X

highest similarity. When the user typed a word, the application will compare the word with all words contained in the smartphone's local database.

Fig.13 Related words

Testing

Testing is done using two test cases, the first test case uses words/sentences typed by the user without selecting words provided by the application. The second test case is using words selected by the user based on the suggestions provided by the application. The accuracy formula used is the calculation of the correct value divided by the total amount of data tested from both test cases. of the 50 words tested in the first case obtained the number of correct words as much as 39 data, while in the second test case used 50 words tested obtained as much as 43 correct transliteration data. The accuracy values for each case are shown in Table 2.

Table 2. Accuracy value for each case study

Case	Total data	False	True	Accuracy value
Case 1	50 words	11	39	70%
Case 2	50 words	7	43	86%
Total	**100 words**	**18**	**82**	**82%**

DISCUSSIONS

The Levenshtein Distance method is used in the app to compare user-typed words with words in the smartphone's local database. By setting a threshold of 50%, the app is able to identify and display the top 10 words that have the highest similarity to the user's typed word. This helps improve the user experience by providing relevant word suggestions and possibly accurate transcription to Balinese script. In the context of this application, data synchronization between the application's internal database and the cloud database plays an important role. The data in the cloud database may be the latest updates or additional new words that can be accessed by the user in real-time. This ensures that the app always has the latest access to transliteration and word translation data.

The results showed that the developed application successfully transliterated with a fairly high accuracy. In the first test, the app was able to transliterate 39 out of 50 words correctly (70% accuracy), while in the second test, where the user selected words from the app's suggestions, the app successfully transliterated 43 out of 50 words correctly (86% accuracy). The total accuracy of both tests was 82%.

*name of corresponding author

This high accuracy rate shows that the FSA method is effective in managing the transliteration behaviour of Latin letters to Balinese script according to the complex writing rules (Astuti et al., 2022). In addition, the 'related words' feature that uses the Levenshtein Distance method allows users to know more about Balinese script, not just limited to the word they are typing. This method has been proven to have a high level of accuracy in finding word similarity and relevance (Rosmala & Risyad, 2018b)(Azhri et al., 2019). This feature helps to increase the user's knowledge of Balinese script and enriches the user's experience in using Balinese script.

Application testing was conducted through two different test cases. The first test case tested the app's ability to transliterate words/sentences typed by the user without selecting words from the app's suggestions. The results showed that of the 50 words tested, 39 were correctly transliterated. The second test case tested the app's ability when the user selects a word from the suggestions given by the app. The result shows that out of 50 words tested, 43 transliteration data are correct. The accuracy of the app is measured based on the number of correct words in both test cases. From the test results, it can be seen that the second test case, where the user selects a word from the app's suggestions, gives slightly higher results compared to the first test case. This shows that the app's word suggestions can help improve the accuracy of the transliteration performed by the user. One of the weaknesses is the transliteration of words that use hangers, where during the transliteration there is a detection error made by the application.

CONCLUSION

In this study, it can be concluded that the use of the Levenshtein Distance method and the implementation of the word suggestion feature in the transliteration application to Balinese script can improve the quality of transliteration performed by users. The tests conducted show that the application has a fairly high accuracy rate in transliterating words from Latin to Balinese script, especially when users choose words from the suggestions given.

ACKNOWLEDGMENT

This work was supported by the Indonesian Ministry of Education and Culture through the Research Grant PDP No. 110/E5/PG.02.00.PL/2024, 2927/LL8/AL.04/2024, 014/INSTIKI.R4.D1/PM.03/06.2024.

REFERENCES

Astuti, D., Kusuma, A. F. A. A., & Syah, F. (2022). Application of the Finite State Machine Method to Determine the End of the Story Based on User Choice in Multiple Role Playing Games. *International Journal of Computer and Information System (IJCIS)*, *3*(2). https://doi.org/10.29040/ijcis.v3i2.70

Aung, K. M. M. (2019). *Comparison of levenshtein distance algorithm and needleman-wunsch distance algorithm for string matching*. MERAL Portal.

Azhri, M. F., Swanjaya, D., & Niswatin, R. K. (2019). Penerapan Algoritma Levenshtein Distance pada Aplikasi Asisten Guru Bahasa Inggris. *Seminar Nasional Inovasi Teknologi*, 155–160.

Belkhir, A., Abdellatif, M., Tighilt, R., Moha, N., Gueheneuc, Y. G., & Beaudry, E. (2019). An observational study on the state of REST API uses in android mobile applications. *Proceedings - 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2019*. https://doi.org/10.1109/MOBILESoft.2019.00020

Birawidya, C. O., Indrawan, G., & Gunadi, I. G. A. (2022). PENGEMBANGAN TRANSLITERASI TEKS AKSARA BALI KE LATIN MENGGUNAKAN FINITE STATE MACHINE. *JURNAL ILMU KOMPUTER INDONESIA*, *7*(1).

*name of corresponding author

Chowdhury, S. D., Bhattacharya, U., & Parui, S. K. (2013). Online handwriting recognition using levenshtein distance metric. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*. https://doi.org/10.1109/ICDAR.2013.24

Devi, N. P. L., & Sudarma, I. K. (2023). Improving Student Balinese Language Learning Outcomes through Interactive Animated Video Based on a Contextual Method. *Journal of Education Technology*, *7*(2).

Errattahi, R., El Hannani, A., & Ouahmane, H. (2018). Automatic speech recognition errors detection and correction: A review. *Procedia Computer Science*, *128*, 32–37.

Gold, C., & Zesch, T. (2020). Exploring the impact of handwriting recognition on the automated scoring of handwritten student answers. *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 252–257.

Hládek, D., Staš, J., & Pleva, M. (2020). Survey of automatic spelling correction. *Electronics*, *9*(10), 1670.

Karmakar, R., Jana, S. S., & Chattopadhyay, S. (2019). A cellular automata guided obfuscation strategy for finite-state-machine synthesis. *Proceedings of the 56th Annual Design Automation Conference 2019*, 1–6.

Mahe, E., Bannour, B., Gaston, C., Lapitre, A., & Le Gall, P. (2024). Finite Automata synthesis from interactions. *Proceedings of the 2024 IEEE/ACM 12th International Conference on Formal Methods in Software Engineering (FormaliSE)*, 12–22.

Noto Sans Balinese. (n.d.).

Prabhakar, D. K., & Pal, S. (2018). Machine transliteration and transliterated text retrieval: a survey. *Sādhanā*, *43*(6), 93.

Rachman, F. H., Qudsiyah, & Solihin, F. (2020). Finite State Automata Approach for Text to Speech Translation System in Indonesian-Madurese Language. *Journal of Physics: Conference Series*, *1569*(2). https://doi.org/10.1088/1742-6596/1569/2/022091

Rosmala, D., & Risyad, Z. M. (2018a). Algoritma Levenshtein Distance dalam Aplikasi Pencarian isu di Kota Bandung pada Twitter. *MIND (Multimedia Artificial Intelligent Networking Database) Journal*, *2*(2), 1–12.

Rosmala, D., & Risyad, Z. M. (2018b). Algoritma Levenshtein Distance dalam Aplikasi Pencarian isu di Kota Bandung pada Twitter. *MIND Journal*, *2*(2), 1–12. https://doi.org/10.26760/mindjournal.v2i2.1-12

Standard, T. U. (2023). □ □. 1–4.

Studi, P., Informatika, T., & Barat, K. (2011). *TELAAH TEORITIS FINITE STATE AUTOMATA DENGAN PENGUJIAN HASIL PADA MESIN OTOMATA*. *1*(1), 59–67.

Utari, N. W. D. A. D., & Ristiani, A. K. (2016). Gumi Bali: Upaya Menghidupkan Bahasa Bali Pada Generasi Muda Ni. *Correspondencias & Análisis*, *15018*, 1–23.

Widiantana, I. K. (2022). MEDIA MASSA BERBAHASA BALI SEBAGAI MEDIA PEMERTAHANAN BAHASA BALI DI PROVINSI BALI: MEDIA MASSA BERBAHASA BALI SEBAGAI MEDIA PEMERTAHANAN BAHASA BALI DI PROVINSI BALI. *Widya Duta: Jurnal Ilmiah Ilmu Sosial Budaya*, *17*(1), 1–9.

*name of corresponding author