A Method for the Affixed Word Transliteration to the Balinese Script on the Learning Web Application

G. Indrawan¹, I K. Paramarta², D. P. Ramendra³, I G. A. Gunadi¹, Sariyasa¹
¹Computer Science Department
²Balinese Language Education Department
³English Language Education Department
Universitas Pendidikan Ganesha (Undiksha), Singaraja, Bali, Indonesia 81116

Abstract: This research proposed a method for the affixed word transliteration to the Balinese Script since there has not been studied yet and it is important since the affixed word needs to be transliterated, inevitably. This research is one of the efforts to preserve digitally the endangered Balinese local language knowledge in Indonesia through the multi-discipline collaboration between Computer Science and Language discipline. The proposed method was taken care of two related aspects, i.e.: (1) A Latin root word has its related Balinese Script root word by using default or special transliteration rule; and (2) A Latin root word with a special transliteration rule for its Balinese Script root word, also need a special transliteration rule for its affixed word. This study was conducted on the pioneering web-based transliteration learning application, BaliScript, that receives the Latin text input and outputs the Balinese Script by using the Noto Serif Balinese font with its dedicated Balinese Unicode. Through the experiment, the proposed method gave the expected transliteration results, added a certain perspective, and strengthened the transliteration knowledge. Future work is to enhance and reuse this method on the mobile computing device, as a part of the Balinese Language ubiquitous learning that supports Balinese Language education, which is a mandatory local subject from the elementary school to the high school in Bali Province.

Keywords: affixed word, Balinese Script, transliteration, learning web

1. Introduction

The Balinese Script as an endangered local language (G. Indrawan, Paramarta, Agustini, & Sariyasa, 2018) in Indonesia affects its Latin-to-Balinese Script transliteration knowledge, as part of the transliteration research in general (Karimi, Scholer, & Turpin, 2011; Kaur & Singh, 2014). The preservation effort by the Bali Government has already been conducted through the Bali Governor Circular Letter (Bali Government, 2018) that follows up the Bali Governor Regulation (Bali Government, 2019). Also, from the previous preservation regulation, the Balinese Language (including all aspects of the Balinese Script) was considered and running as a mandatory local subject from elementary school to senior high school in Bali Province.

As the preservation effort of the language can be done by that governmental/political approach, multiple approaches should strengthen it and should have a greater impact. This research joined the effort through the technological approach, through the multi-discipline collaboration between Computer Science and Language discipline, by proposing a method for the affixed word transliteration to the Balinese Script since there has not been studied yet and it is important since the affixed word needs to be transliterated, inevitably. Moreover, this study was conducted on the pioneering web-based transliteration learning application, BaliScript, as part of the planned Balinese Language ubiquitous learning that supports Balinese Language education. The proposed method can be reused in the future for the mobile application as part of ubiquitous learning to give the transliteration knowledge for the next generations to come. As known, ubiquitous learning improves learning by providing learners with content and interaction anytime and anywhere (Hwang, Tsai, & Yang, 2008) by mobile and embedded computers and wireless networks in everyday life (Ogata, Matsuoka, Bishouty, & Yano, 2009).

This research proposes an unexposed method for the affixed word transliteration to the Balinese Script based on the supporting computer font Noto Serif Balinese (NSB). Also, this work advances the previous authors’ work on the transliteration process by: (1) accommodating the rule from the Balinese Language, Script, and Literature Advisory Agency (Anom et al., 2009). This Bali Province government agency (Bali Government, 1992) carries out guidance and formulates programs for the maintenance, study, development, and preservation of the Balinese Language, Script, and Literature; (2) accommodating seventeen types of special words (G. Indrawan, Paramarta, & Agustini, 2019; G. Indrawan, Paramarta, et al., 2018) by using a particular table structure in the database which is avoiding hard-coded special word repositories in the learning application; (3) utilizing the Noto Serif Balinese font (Google, 2020b; The Unicode Consortium, 2020b, 2020a) with more developed and less bug than the Noto Sans Balinese font (Google, 2020a). Also, it uses dedicated Balinese
Unicode as a standard Balinese Script font on the computer system including mobile devices (so the method can be reused on the mobile application); and (4) maximizing the learning application, that used this method, for the delivery of the knowledge in the area of transliteration and also translation, at the same time, specifically for the Balinese Script, Indonesian, and English (see the next Figure 2). Overall, all of those advances were considered as the contribution of this work.

This paper is organized into several sections, i.e.: Section 1 (Introduction) states the problem background related to the Balinese Script in general, and the Latin-to-Balinese Script transliteration in specific; Section 2 (Related Work) describes the related works in the area of the Latin-to-Balinese Script transliteration; Section 3 (Research Method) exposes the supporting transliteration computer font (which is the NSB font) with its dedicated Balinese Unicode Unicode, and the supporting rule-based algorithm for the affixed word transliteration to the Balinese Script; Section 4 (Result and Discussion) covers the analysis of the testing result; and finally, Section 5 (Conclusion and Future Work) consists of some important conclusion and future work points.

2. Related Work

As the Balinese Script adheres with its scriptio-continua writing style like its relatively close (geographically and culturally) Javanese Script (Widiarti & Pulungan, 2020), several related works on Latin-to-Balinese Script transliteration were conducted on the previous author’s works (Crisnapati et al., 2019; G. Indrawan, Dantes, Aryan, & Paramarta, 2020; G. Indrawan, Gunadi, Gitakarma, & Paramarta, 2021; G. Indrawan, Paramarta, et al., 2019; G. Indrawan, Puspita, Paramarta, & Sariyasa, 2018; G. Indrawan, Sariyasa, & Paramarta, 2019; G. Indrawan, Setemen, Sutaya, & Paramarta, 2020; G. Indrawan, Swastika, Sariyasa, & Paramarta, 2020; Loekito, Indrawan, Sariyasa, & Paramarta, 2020). Various non-dedicated Balinese Unicode fonts (i.e. Bali Simbar (Suatjana, 1999) and Bali Simbar Dwijendra (Suatjana, 2009)) and dedicated Balinese Unicode font (The Unicode Consortium, 2020b, 2020a) (i.e. Noto Sans Balinese (Google, 2020a)) were used for displaying Balinese Script output on those previous research. A more developed dedicated Balinese Unicode font was found on Noto Serif Balinese font (Google, 2020b) rather than Noto Sans Balinese font. Bali Simbar font was used in (G. Indrawan, Sariyasa, et al., 2019) with a relatively good accuracy result on testing cases from The Balinese Alphabet document (Sudewa, 2003). It was also used on the developed robotic system that writing Balinese Script from Latin text input (Crisnapati et al., 2019), and on the transliteration line-breaking handling exploration (G. Indrawan, Setemen, et al., 2020). Bali Simbar Dwijendra (BSD) font, as the improvement of Bali Simbar (BS) font, was used in (G. Indrawan, Swastika, et al., 2020) with additional testing cases from the Balinese Script dictionary (Anom et al., 2009), in addition to the same testing cases on (G. Indrawan, Sariyasa, et al., 2019). It was also used on the transliteration exploration of the mathematical expression (G. Indrawan, Dantes, et al., 2020). Ten transliteration lessons were also learned by using this font on the extended testing data (G. Indrawan et al., 2021) other than the initial established testing data (G. Indrawan, Paramarta, et al., 2019; G. Indrawan, Sariyasa, et al., 2019), Noto Sans Balinese font was used in (G. Indrawan, Paramarta, et al., 2019) with the same testing cases in (G. Indrawan, Sariyasa, et al., 2019) and gave a relatively good accuracy result. It was also used on the developed robotic system that writing Balinese Script from Latin text input (G. Indrawan, Puspita, et al., 2018). Extensive accuracy analysis on the developed algorithm (G. Indrawan, Paramarta, et al., 2019) was conducted in (Loekito et al., 2020) for improvement in the future.

For Balinese Script preservation effort and learning, both sides of transliteration knowledge need to be explored from a technological point of view for digitalization. Ref. (G. Indrawan, Ariawan, Agustini, & Paramarta, 2020) was related to the Balinese Script-to-Latin transliteration along with the other authors’ work (Gede Indrawan, Gunadi, & Paramarta, 2020) that utilizing GNU Optical Character Recognition (OCR), i.e. Ocrad (Antonio Diaz, 2003). Ref. (Gede Indrawan et al., 2020) is still limited only to basic syllable (Sudewa, 2003) recognition from the Balinese Script image that was based on Bali Simbar glyph shape. For advancing functionality and mobile adoption for ubiquitous learning, utilization of Tesseract OCR was conducted by (G. Indrawan, Ariawan, et al., 2020).

3. Research Method

The proposed method for the affixed word transliteration to the Balinese Script was relied on two related aspects, i.e.: (1) A Latin root word has its related Balinese Script root word by using default or special transliteration rule; and (2) A Latin root word with a special transliteration rule for its Balinese Script root word, also need a special transliteration rule for its affixed word.

This section describes: (1) the supporting transliteration computer font (which is the NSB font) with its dedicated Balinese Unicode, as a standard Balinese Script font on the computer system including mobile devices; and (2) the supporting rule-based algorithm for the affixed word transliteration to the Balinese Script.

1 https://github.com/tesseract-ocr/ (Retrieved April 25, 2021)
3.1. The Supporting Computer Font

The proposed method for the affixed word transliteration to the Balinese Script involves the Noto Serif Balinese (NSB) font with its dedicated Balinese Unicode (Google, 2020b; The Unicode Consortium, 2020b, 2020a), as shown by Table 1. NSB font accomodates Balinese Script Complex Behaviours (Narendra, 2008) which cover: (1) reordering and splitting; (2) various placement of diacritics; (3) contextual shaping; (4) complex ligature construction.

Table 1 shows 121 glyphs and 7 code points in reserve grey areas. The Unicode code point group allocation is at U+1B00 – U+1B7F, i.e.: (1) 1B00–1B04 for various signs; (2) 1B05–1B12 for independent vowels; (3) 1B13–1B33 for consonants; (4) 1B34 for sign rerekan; (5) 1B35–1B43 for dependent vowel signs; (6) 1B44 for sign adeg–adeq; (7) 1B45–1B4B for additional consonants (Aksara Sasak); (8) 1B50–1B59 for digits; (9) 1B5A–1B60 for punctuation; (10) 1B61–1B6A for musical symbols for notes; (11) 1B6B–1B73 for diacritical marks for musical symbols; and (12) 1B74–1B7C for musical symbols.

Table 1. Glyphs and their related Unicode of the Noto Serif Balinese font

| U+1B01 | U+1B02 | U+1B03 | U+1B04 | U+1B05 | U+1B06 | U+1B07 | U+1B08 | U+1B09 | U+1B0A | U+1B0B | U+1B0C | U+1B0D | U+1B0E | U+1B0F |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0      | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | A      | B      | C      | D      | E      |

3.2. The Algorithm

The algorithm for the affixed word transliteration to the Balinese Script was based on two related aspects, as described previously. For an example of the first aspect, the root word “aksara” (alphabet) (Sudewa, 2003) must be transliterated to the Balinese Script as “aksara” (special transliteration), not “aksara” (default transliteration). The vowel “a” must be transliterated to “a” (Balinese Letter Akara, U+1B05) not “a” (Balinese Letter Ha, U+1B33), and the consonant-vowel cluster “ksa” must be transliterated to “aksara” (constructed by Balinese Letter Ka, U+1B13, and Balinese Letter Sa Sapa, U+1B31), not “aksara” (constructed by Balinese Letter Ka, U+1B13, and Balinese Letter Sa Sapa, U+1B31) (The Unicode Consortium, 2020a).

For an example of the second aspect, related to the root word “aksara” from the first aspect, its affixed word (in this case, root word with suffix) “aksarane” (that alphabet) (Sudewa, 2003) must be transliterated to the Balinese Script as “aksarane” (special transliteration), not “aksarane” (default transliteration). The syllable “ne” was constructed by “ne” (Balinese Letter Na, U+1B26) and “a” (Balinese Vowel Sign Taling, U+1B3E) (The Unicode Consortium, 2020a).

That example of the affixed word transliteration and several other cases (using prefix and/or suffix) should be handled by the proposed method. Algorithm 1 shows the pseudocode for the affixed word transliteration to the Balinese Script. Figure 1 shows the generated SQL instruction for the example of word_string = “aksarane” with word_string_length = 8.

Based on the example, line 01 gave part of SQL instruction, i.e., “SELECT ‘aksarane’ varian, 8 varianLength”. Line 02 sets word_min = 3 which means searching in the database for the root words with a minimum of three characters. Line 03 sets the initial value of word_substring = “aksara” for the first for iteration at line 04. Line 05 and 06 set the initial value of word_substring2 = “aksarane” with word_substring2_length = 7 for the second for iteration at line 07. Line 08 gave part of SQL instruction, i.e., “UNION ALL SELECT ‘aksaran’, ‘aksarane’”. Line 09 updates the value of word_substring2 = “aksara” with word_substring2_length = 6 for the second for iteration at line 07. The condition of word_substring2 > word_min would break the second for iteration at line 07 and continue with the first for iteration at line 04. Figure 1 shows the complete variants of word_substring2 that would be JOIN with records in the database if column “kata” equal to those variants of word_substring2. Line 13 result was shown at the bottom section of Figure 1. Only the top record was used as the result since it has the biggest match level compare to the other records. For the sake of...
clarity, the other records result was shown by commented line “LIMIT 1” of the constructed SQL instruction.

Algorithm 1. The affixed word transliteration to the Balinese Script

01 \textbf{insert} \textit{word_string} and its \textit{word_string} length to SQL SELECT instruction
02 \textbf{set} \textit{word_min} = 3
03 \textbf{set} \textit{word_substring} = \textit{word_string}
04 \textbf{for} \textit{word_substring} and \textit{word_substring} ≥ \textit{word_min}, do
05 \textbf{set} \textit{word_substring2} = \textit{word_substring}
06 \textbf{update} \textit{word_substring2} by removing its most left character
07 \textbf{for} \textit{word_substring2} and \textit{word_substring2} ≥ \textit{word_min}, do
08 \textbf{insert} \textit{word_substring2} and its \textit{word_substring2} length to SQL UNION ALL SELECT instruction
09 \textbf{update} \textit{word_substring2} by removing its most right character
10 \textbf{end}
11 \textbf{update} \textit{word_substring} by removing its most left character
12 \textbf{end}
13 \textbf{retrieve} \textit{kata} and its related \textit{konvert} from the database by using constructed SQL instruction with “records.is_exclude = 0” and “LIMIT 1”
14 \textbf{update} \textit{word_string} by replacing its same string part (refer to \textit{kata}) with its related \textit{konvert} of \textit{kata}
15 \textbf{go to} the next transliteration process to get the Balinese Script output from the \textit{word_string}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{algorithm1.png}
\caption{The example result from the fully constructed SQL instruction that was generated by the algorithm}
\end{figure}

Finally, line 15 gave \textit{word_string} = “\textasciitildeak\textasciitildearane” from column value “\textit{konvert}” for the next transliteration process to give the right result “ᬅᬓ᭄ᬱᬭᬾ”. As a note, the SQL instruction search in the database for all related words which have flag status \textit{is_excluded} = 0. Also “\textit{konvert}” was referred to as Balinese code in the next
3.3. The Testing

The testing of the proposed method was conducted on the web application that was run on Intel(R) Core(TM) i7-4600U CPU @ 2.09GHz platform with 8 GB RAM and Windows 8 64-bit Operating System. That web application was constructed by the web server of Apache 2.4.46 (Win64), the database server MySQL 10.4.18-MariaDB, and programming language PHP 7.4.16 combined with JavaScript. Also as one of the contributions of this work, this web learning application, that used this method, maximizing the learning experience through the delivery of the knowledge in the area of transliteration and also translation at the same time, specifically for the Balinese Script, Indonesian, and English (Figure 2). The translation information comes from the columns from the database. For example, Indonesian translation for certain transliterated word can be seen in the column “indo” (Figure 1).

Table 2 shows the testing transliteration cases on several affixed words that exist from the testing data set of the previous authors’ work (G. Indrawan, Paramarta, et al., 2019). Those cases were also accompanied by their counterpart testing cases that comply with the rule from the Balinese Language, Script, and Literature Advisory Agency (Anom et al., 2009). All of those testing cases, along with the root words to be transliterated, consists of various affixed words (the bold words) that using a prefix and/or suffix.

| Case¹ | Remarks |
|-------|---------|
| 1 Bank Pembangunan Daerah Bali Be Pe De Bali Ba Pe Da Bali Ba Pa Da Bali | Abbreviations section in (Sudewa, 2003). Its English: Regional Development Bank | |
| 2 Akeh aksaranë, 47, luir ipun: aksara suara, 14, aksara wianjana, 33, aksara suara punika talêr dados pangangge suara, tur madrêwe suara kakalih, kawâstantin: suara hrêswa miwah dirgha. | Line Break section in (Sudewa, 2003). Its English: Many of those letters, 47, as follows vowels, 14, consonants, 33, those vowels also become vowel signs, and have two sounds, that was called: sound hrêswa and dirgha. |

1The original testing cases from the previous authors’ work testing data set (G. Indrawan, Paramarta, et al., 2019)
2The updated testing cases that comply with the rule (Anom et al., 2009)

Figure 2. The BaliScript transliteration learning web application with translation feature for the Balinese Script, Indonesian, and English
4. Result and Analysis

Figure 3 shows the testing result of the proposed method for the affixed word transliteration on the transliteration learning web application BaliScript. The testing on the BaliScript was based on testing cases in Table 2. Figure 3(a) shows the input with various affixed words through the select input box⁵, while Figure 3(b) shows the transliteration output with a line break for each testing case (Table 2). Several aspects were analyzed related to the transliteration process that was successfully displayed by the Noto Serif Balinese (NSB) font with its dedicated Balinese Unicode.

Balinese Script writing uses the non-scriptio-continua style (Widiarti & Pulungan, 2020) which means the output should fully occupy all of the writing space without paragraph and line break. For ease of learning and also for the ease of visual analysis, this aspect could be accommodated in the future by a feature to switch between the scriptio-continua style (the traditional look) and the non-scriptio-continua style (the modern look). The non-scriptio continua style using Cascading Style Sheets (CSS) property white-space⁶ with value pre-line.

![Figure 3](image-url)

**Figure 3.** Transliteration result of various affixed words on the BaliScript learning web application: (a) the input through the select box; (b) the output with a non-scriptio-continua style.

The first affixed word, i.e. an Indonesian word “Pembangunan” (development) was constructed by a prefix “pem”, a root word “bangun” (develop), and a suffix “an”. The writing of “e” and its sound [ə] (Esling, 1999) complies with the rule from the Balinese Language, Script, and Literature Advisory Agency (Anom et al., 2009).

⁵ https://select2.org (Retrieved April 25, 2021)
⁶ https://www.w3schools.com/cssref/pr_text_white-space.asp (Retrieved April 25, 2021)
The affixed word “Pembangunan” was successfully transliterated by the proposed method to the Balinese Script “ᬦᬗᬦᬦ᭄ᬱᬭᬾ” by using its Balinese code (see column “konvert” at the previous algorithm section) related to the existing root word “bangun” (“ᬪᬦ᭄ᬱᬭᬾ”) in the database. As a note, consonant “n” (“ᬦ”) of word “Pembangunan” was transformed to “nda” (“ᬦᬦ”) since it was followed by a syllable “da” from word “Daerah”. The appended form of “da” was positioned under the regular form of “na” (“ᬦ”).

The second affixed word, i.e. a Balinese word “akśarane” (that alphabets) was constructed by a root word “akśara” (alphabets) and a suffix “ne”. The root word “akśara” (Sudewa, 2003), as a variant word of “aksara” (Anom et al., 2009) with the same transliteration, is accommodated by the proposed method. Since its intended sound is [e] (Esling, 1999), to comply with the rule from the Balinese Language, Script, and Literature Advisory Agency (Anom et al., 2009), the writing of suffix “ne” should be changed to “nē”. Based on that change, the affixed word “akśarane” was successfully transliterated by the proposed method to the Balinese Script “ᬦᬫ᭄ᬩᬗ᭄ᬤᬺᬯᬾ” by using its Balinese code related to the existing root word “aksara” (“ᬦᬗ᭄ᬱᬭᬾ”) in the database.

The third affixed word, i.e. a Balinese word “punika” (that) was constructed by a prefix “pu”, and a root word “nika” (that). The affixed word “punika” was successfully transliterated by the proposed method to the Balinese Script “ᬦᬪᬦᬦ᭄ᬱᬭᬾ” by using its Balinese code related to the existing root word “nika” (“ᬦᬦ᭄ᬱᬭᬾ”) in the database.

The fourth affixed word, i.e. a Balinese word “pangangge” (sign) was constructed by a prefix “pang”, and a root word “angge” (use). The root word “angge” (Sudewa, 2003), as a variant word of “anggē” (Anom et al., 2009) with the same transliteration, is accommodated by the BaliScript. Rather than using its Balinese code related to the existing root word “angge” (“ᬪᬵ᭄ᬓᬯᬱ᭄ᬝᬦᬶᬦ᭄ᬵ”) in the database, the affixed word “pangangge” was successfully transliterated by the proposed method to the Balinese Script “ᬗᬗ᭄ᬱᬭᬾ” by using its Balinese code related to the existing root word “ngangge” (“ᬗᬗ᭄ᬱᬭᬾ”) in the database (see Algorithm 1). As a note, syllable “pa” (“ᬦ”) of word “pangangge” was transformed to “spa” (“ᬦᬦ”) since it followed a consonant “s” (“ᬦ”) from word “dados”. The appended form of “pa” was positioned after the regular form of “sa” (“ᬦ”).

The fifth affixed word, i.e. a Balinese word “madrēwe” (have/has) was constructed by a prefix “ma” and a root word “drēwe” (belonging). The root word “drēwe” (Sudewa, 2003), as a variant word of “drewē” (Anom et al., 2009) with the same transliteration, is accommodated by the BaliScript. The affixed word “madrēwe” was successfully transliterated by the proposed method to the Balinese Script “ᬦᬪᬦᬦ᭄ᬱᬭᬾ” by using its Balinese code related to the existing root word “drēwe” (“ᬦᬗ᭄ᬱᬭᬾ”) in the database.

The sixth affixed word, i.e. a Balinese word “kakalih” (both) was constructed by a prefix “ka” and a root word “kali” (two). Rather than using its Balinese code related to the existing root word “kali” in the database, the affixed word “kakalih” was successfully transliterated by the proposed method to the Balinese Script “ᬦᬦ᭄ᬱᬭᬾ” by setting the flag status “is_excluded” = 1 of the existing root word “kali” (“ᬦᬦ᭄ᬱᬭᬾ”) in the database (see Algorithm 1). As a note, rather than using the Balinese Script “ᬦᬦ᭄ᬱᬭᬾ” (Sudewa, 2003) as the ground truth for the transliteration result of the affixed word “kakalih”, the proposed method used “ᬦᬦ᭄ᬱᬭᬾ” (with “จอง”) as the Balinese Vowel Sign Tedung, U+1B35) since it complies with the rule from the Balinese Language, Script, and Literature Advisory Agency (Anom et al., 2009).

The seventh affixed word, i.e. a Balinese word “kawāśţanin” (be named) was constructed by a prefix “ka”, a root word “wāśţa” (name), and a suffix “nin”. The root word “wāśţa” (Sudewa, 2003), as a variant word of “wasta” (Anom et al., 2009) with the same transliteration, is accommodated by the Balinese Script. The affixed word “kawāśţanin” was successfully transliterated by the proposed method to the Balinese Script “ᬦᬫ᭄ᬩᬗ᭄ᬤᬺᬯᬾ” by using its Balinese code related to the existing root word “wāśţa” (“ᬦᬦ᭄ᬱᬭᬾ”) in the database. As a note, rather than using the Balinese Script “ᬦᬫ᭄ᬩᬗ᭄ᬤᬺᬯᬾ” (Sudewa, 2003) as the ground truth for the transliteration result of the affixed word “kakalih”, the proposed method used “ᬦᬦ᭄ᬱᬭᬾ” (without “จอง”) as the Balinese Vowel Sign Tedung.
Sign Tedung, U+1B35) since it complies with the rule from the Balinese Language, Script, and Literature Advisory Agency (Anom et al., 2009).

**Conclusion and Future Work**

A method for the affixed word transliteration to the Balinese Script was proposed with the implementation on the web learning application by using Noto Serif Balinese font with its dedicated Balinese Unicode. All various cases were handled successfully by the proposed method. This research added a certain perspective and strengthened the transliteration knowledge, which is part of the Balinese culture preservation effort.

Future work to enrich and strengthen this transliteration knowledge is to enhance and reuse this method on mobile computing through the implementation on the mobile device, as a part of the planned Balinese Language ubiquitous learning that supports Balinese Language education.

**Acknowledgment**

This work was supported by the Indonesian Ministry of Education, Culture, Research, and Technology through the multi-years Applied Research Grant. The application used by this research has also been approved by the Indonesian Ministry of Law and Human Rights for the copyright certificate with No. 000235267.

**References**

1. Anom, I. G. K., Suasta, I. B. M., Suardiana, I. W., Djapa, I. W., Suteja, I. W., Riken, I. M., & Swatjana, I. M. (2009). *Balinese - Indonesian Dictionary with its Latin and Balinese Script* [Kamus Bali - Indonesia Beraksara Latin dan Bali]. Denpasar: Bali Province.
2. Antonio Díaz. (2003). Ocrad - The GNU OCR. Retrieved April 25, 2021, from https://www.gnu.org/software/ocrad/
3. Bali Government. (1992). Bali Government Regulation No. 3 on Balinese Language, Script, and Literature [Peraturan Pemerintah Bali No. 3 tentang Bahasa, Aksara, dan Sastra Bali]. Retrieved April 25, 2021, from https://bphn.go.id/data/documents/perda_3_1992.pdf
4. Bali Government. (2018). Bali Governor Regulation No. 80 on Protection and Usage of Balinese Language, Script, and Literature, also Organizing Balinese Language Month [Peraturan Gubernur Bali No. 80 tentang Pelindungan dan Penggunaan Bahasa, Aksara, dan Sastra Bali serta Penyelenggaraan Bulan Bahasa Bali]. Retrieved April 25, 2021, from https://jdih.baliprov.go.id/produkhukum/peraturan/abstrak/24665
5. Bali Government. (2019). Bali Governor Circular Letter No. 3172 Year 2019 about The Usage of Balinese Traditional Clothing and Balinese Script [Surat Edaran Gubernur Bali No. 3172 Tahun 2019 tentang Penggunaan Busana Adat Bali dan Aksara Bali]. Retrieved April 25, 2021, from https://jdih.baliprov.go.id/produkhukum/peraturan/abstrak/24741
6. Crisnapati, P. N., Indrawan, G., Novayanti, P. D., Suranata, I. W. A., Pande, I. M. S. A., Yusadara, I. G. P. M., & Santrisna, I. P. (2019). Pasang Aksara Bot: A Balinese Script Writing Robot using Finite State Automata Transliteration Method. *Journal of Physics: Conference Series*, 1175(1), 012108. https://doi.org/10.1088/1742-6596/1175/1/012108
7. Esling, J. (1999). *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press.
8. Google. (2020a). Google Noto Fonts. Retrieved April 25, 2021, from https://www.google.com/get/noto/#sans-bali
9. Google. (2020b). Google Noto Serif Balinese. Retrieved April 25, 2021, from https://github.com/googlefonts/noto-fonts/blob/master/unhinted/ttf/NotoSerifBalinese/NotoSerifBalinese-Regular.ttf
10. Hwang, G. J., Tsai, C. C., & Yang, S. J. H. (2008). Criteria, strategies and research issues of context-aware ubiquitous learning. *Journal of Educational Technology and Society*, 11(2). https://doi.org/10.1168/s40561-014-0004-5
11. Indrawan, G., Ariawan, K. U., Agustini, K., & Paramarta, I. K. (2020). Finite-State Machine for Post-Processing Method of Balinese Script to Latin Transliteration. In *The 6th International Conference on Science, Technology, and Interdisciplinary Research*. In press
12. Indrawan, G., Dantes, G. R., Aryanto, K. Y. E., & Paramarta, I. K. (2020). Handling of Mathematical Expression on Latin-to-Balinese Script Transliteration Method on Mobile Computing. *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 1–5. IEEE. https://doi.org/10.1109/ICIC50835.2020.9288563
13. Indrawan, G., Gunadi, I. G. A., Gitakarma, M. S., & Paramarta, I. K. (2021). Latin to Balinese Script Transliteration: Lessons Learned from Computer-based Implementation. In *The 4th International Conference on Software Engineering and Information Management (ICSIM)*. Yokohama. In press
14. Indrawan, G., Paramarta, I. K., & Agustini, K. (2019). A New Method of Latin-to-Balinese Script Transliteration based on Noto Sans Balinese Font and Dictionary Data Structure. The 2nd International Conference on Software Engineering and Information Management (ICSIM). Singaraja. https://doi.org/10.1145/3305160.3305167

15. Indrawan, G., Paramarta, I. K., Agustini, K., & Sariyasa. (2018). Latin-to-Balinese Script Transliteration Method on Mobile Application: A Comparison. Indonesian Journal of Electrical Engineering and Computer Science, 10(3), 1331. https://doi.org/10.11591/ijeecs.v10.i3.pp1331-1342

16. Indrawan, G., Puspita, N. N. H., Paramarta, I. K., & Sariyasa. (2018). LBtrans-Bot: A Latin-to-Balinese Script Transliteration Robotic System based on Noto Sans Balinese Font. Indonesian Journal of Electrical Engineering and Computer Science, 12(3), 1247. https://doi.org/10.11591/ijeecs.v12.i3.pp1247-1256

17. Indrawan, G., Sariyasa, & Paramarta, I. K. (2019). A New Method of Latin-To-Balinese Script Transliteration based on Bali Simbar Font. 2019 Fourth International Conference on Informatics and Computing (ICIC), 1–6. Semarang: IEEE. https://doi.org/10.1109/ICIC47613.2019.8985675

18. Indrawan, G., Setemen, K., Sutaya, W., & Paramarta, I. K. (2020). Handling of Line Breaking on Latin-to-Balinese Script Transliteration Web Application as Part of Balinese Language Ubiquitous Learning. The 6th International Conference on Science in Information Technology (ICSItech). Singaraja. https://doi.org/10.1109/ICSI Tech49800.2020.9392035

19. Indrawan, G., Swastika, I. P. E., Sariyasa, & Paramarta, I. K. (2020). An Improved Algorithm and Accuracy Analysis Testing Cases of Latin-to-Balinese Script Transliteration Method based on Bali Simbar Dwijendra Font. Test Engineering and Management, 83, 7676–7683. Retrieved from http://www.testmagazine.biz/index.php/testmagazine/article/view/5056

20. Indrawan, Gede, Gunadi, I. G. A., & Paramarta, I. K. (2020). Towards Ubiquitous Learning of Balinese-to-Latin Script Transliteration as Part of Balinese Language Education. Proceedings of the 4th Asian Education Symposium (AES 2019). Paris, France: Atlantis Press. https://doi.org/10.2991/assehr.k.200513.025

21. Karimi, S., Scholer, F., & Turpin, A. (2011). Machine transliteration survey. ACM Computing Surveys, 43(3), 1–46. https://doi.org/10.1145/1922649.1922654

22. Kaur, K., & Singh, P. (2014). Review of Machine Transliteration Techniques. International Journal of Computer Applications, 107(20), 13–16. https://doi.org/10.5120/18866-0061

23. Loekito, L. H., Indrawan, G., Sariyasa, & Paramarta, I. K. (2020). Error Analysis of Latin-to-Balinese Script Transliteration Method Based on Noto Sans Balinese Font. Proceedings of the 3rd International Conference on Innovative Research Across Disciplines (ICIRAD 2019). Paris, France: Atlantis Press. https://doi.org/10.2991/assehr.k.200115.055

24. Narendra, B. B. (2008). The Development of Balinese Script Smart Font [Pembuatan Smart Font Aksara Bali], Undergraduate Thesis. Institut Teknologi Bandung, Bandung.

25. Ogata, H., Matsuka, Y., Bishouty, M. M. El, & Yano, Y. (2009). LORAMS: linking physical objects and videos for capturing and sharing learning experiences towards ubiquitous learning. International Journal of Mobile Learning and Organisation, 3(4), 337. https://doi.org/10.1504/IJMLO.2009.027452

26. Suatjana, I. M. (1999). Bali Simbar. Retrieved April 25, 2021, from http://www.babadbali.com/aksarabali/balisimbar.htm

27. Suatjana, I. M. (2009). Bali Simbar Dwijendra. Denpasar: Yayasan Dwijendra.

28. Sudewa, I. B. A. (2003). The Balinese Alphabet. Retrieved April 25, 2021, from http://www.babadbali.com/aksarabali/alphabet.htm

29. The Unicode Consortium. (2020a). Balinese Unicode Table. Retrieved April 25, 2021, from http://unicode.org/charts/PDF/UC1BB.pdf

30. The Unicode Consortium. (2020b). The Unicode Standard Version 13.0 – Core Specification (The Unicode Consortium, Ed.). The Unicode Consortium. Retrieved from http://www.unicode.org/versions/Unicode13.0.0/

31. Widiarti, A. R., & Pulungan, R. (2020). A method for solving scriptio continua in Javanese manuscript transliteration. Heliyon, 6(4), e03827. https://doi.org/10.1016/j.heliyon.2020.e03827