ICIC Express Letters
Volume **16**, Number **7**, July **2022**

ICIC International ©2022 ISSN 1881-803X
pp. **687–694**

A METHOD FOR NON-ALPHANUMERIC TEXT PROCESSING ON TRANSLITERATION TO THE BALINESE SCRIPT

Gede Indrawan[1,*], I Wayan Sutaya[1], Ketut Udy Ariawan[1]
Made Santo Gitakarma[1], I Gede Nurhayata[1] and I Ketut Paramarta[2]

[1]Faculty of Engineering and Vocational
[2]Faculty of Language and Art
Universitas Pendidikan Ganesha
Jalan Udayana 11, Singaraja, Bali 81116, Indonesia
{ wsutaya; udyariawan; santo; gede.nurhayata; ketut.paramarta }@undiksha.ac.id
*Corresponding author: gindrawan@undiksha.ac.id

Received July 2021; accepted October 2021

Abstract. *This research proposed a method for non-alphanumeric text processing on transliteration to the Balinese Script. The objective is to know the transliteration handling of this kind of input text comprehensively. It is considered as the main contribution of this work since it has not been studied yet. This multi-discipline collaboration work is part of the technological efforts to preserve the endangered Balinese local language knowledge in Indonesia. The proposed method used the dedicated Balinese Unicode font to display the transliteration result through the pioneering web-based learning application, BaliScript, that receives the text input and outputs the Balinese Script. There are three aspects related, i.e., 1) Balinese Language traditionally uses the phrase approach to represent the non-alphanumeric expressions in its Balinese writing; 2) In the computer-based transliteration implementation, to ease learning, non-alphanumeric expressions should be preserved to avoid complexity related to their various phrase expressions with the same meaning; 3) The second aspect was limited by the supporting computer font and needed a special algorithm. Through the experiment, the proposed method gave good transliteration results related to these non-alphanumeric expressions and added a certain perspective despite the inherent limitation of the previously stated third aspect.*
Keywords: Balinese Script, Non-alphanumeric text, Transliteration, Unicode font

1. **Introduction.** As part of the transliteration research in general [1,2], Balinese Script transliteration knowledge raises preservation concerns. It is part of the endangered local Balinese Language in Indonesia and being a subject of preservation efforts through the Bali Governor Regulation[1,2] and the Bali Governor Circular Letter[3]. These efforts make the Balinese Language running as a mandatory local subject from basic school to senior high school in Bali Province. To strengthen the preservation efforts, multiple approaches should have a greater impact. This research joined the effort through the technological approach, by proposing a method for non-alphanumeric text processing on transliteration to the Balinese Script. The objective is to know the transliteration handling of this kind of input text comprehensively. It is considered as the main contribution of this work since it has not been studied yet.

This study used the pioneering web-based transliteration learning application, BaliScript, and the proposed method is reusable on the mobile application for further ubiquitous

DOI: 10.24507/icicel.16.07.687

[1]Bali Government Regulation No. 3 Year 1992 on Balinese Language, Script, and Literature

[2]Bali Governor Regulation No. 80 Year 2018 on Protection and Usage of Balinese Language, Script, and Literature

[3]Bali Governor Circular Letter No. 3172 Year 2019 about The Usage of Balinese Traditional Clothing and Balinese Script

Balinese Language learning [3,4]. This work is relatively close to and advances the previous work [5] on the Balinese transliteration process by 1) complying with the rule from the Balinese Language, Script, and Literature Advisory Agency [6]; 2) accommodating special words [8,9] in the database's table then avoiding hard-coding them in the learning application; 3) using Noto Serif Balinese font[4,5] [9] to display the Balinese Script as the transliteration result. This font uses dedicated Balinese Unicode as a standard Balinese Script font on the computer system including mobile devices. This is a factor that makes the proposed method can be reused on a mobile application, as described previously; 4) enhancing the learning experience on the application (that used the proposed method) by featuring the Indonesian and English translation for the transliterated word [10]. Overall, all of those advances are considered as the contribution of this work.

This paper is organized into several sections, i.e., Section 1 (Introduction) conveys the problem background related to the Balinese Script in general, and the transliteration to Balinese Script in specific; Section 2 (Related Works) describes the related works in this research area; Section 3 (State of the Art) covers the supporting non-alphanumeric glyphs, the algorithm, and the testing; Section 4 (Result and Analysis) gives the analysis of the testing result; Section 5 (Conclusions) consists of some important conclusion points.

2. **Related Works.** Researches on transliteration to Balinese Script were conducted on the previous works [5,8,10-17]. Balinese fonts (i.e., Bali Simbar[6], Bali Simbar Dwijendra [18], Noto Sans Balinese[7], and Noto Serif Balinese [9]) were used to display the Balinese Script output. Bali Simbar (BS) font was used in [11] with a relatively good accuracy result on testing cases from The Balinese Alphabet document[8]. The robotic system of writing Balinese Script from text input [12] and exploration on line-breaking handling of transliteration [13] also used this font. Bali Simbar Dwijendra (BSD) font, as the improvement of BS font, was used in [14] with additional testing cases from the Balinese Script dictionary [6], in addition to the same testing cases on [11]. The transliteration exploration of the mathematical expression [5] and ten learned lessons on computer-based transliteration [15] also used this font. Noto Sans Balinese font was used in [8] with the same testing cases in [11] and gave a relatively good accuracy result. The robotic system of writing Balinese Script from text input [16] and extensive accuracy analysis on the developed algorithm [17] also used this font. More developed and less bug of Noto Serif Balinese font was used in the study of backward compatibility transliteration [10].

All of those related works were related to alphanumeric text processing. None of them exposed comprehensively its counterpart non-alphanumeric text processing on transliteration to the Balinese Script. This becomes the motivation of this paper besides the enhancement to the previous work [5], as described previously.

3. **State of the Art.** The proposed method for processing of non-alphanumeric text (including punctuation, mathematical, and general symbols) on transliteration to the Balinese Script relied on three related aspects, i.e., 1) Balinese Language traditionally uses the phrase approach rather than the symbol approach (like "−", and "=") to represent the non-alphanumeric expressions in its Balinese writing. For example, "$1 - 1 = 0$" is written based on the phrase approach, like "besik kuang besik patuh teken telah" (one subtracted by one equal to zero). Several variants of this approach exist with the same meaning, like

[4]Balinese Unicode Table, http://unicode.org/charts/PDF/U1B00.pdf (Retrieved on August 17, 2021)

[5]Google Noto Serif Balinese, https://github.com/googlefonts/noto-fonts/blob/master/unhinted/ttf/NotoSerifBalinese/NotoSerifBalinese-Regular.ttf (Retrieved on August 17, 2021)

[6]Bali Simbar, http://www.babadbali.com/aksarabali/balisimbar.htm (Retrieved on August 17, 2021)

[7]Google Noto Fonts, https://www.google.com/get/noto/#sans-bali (Retrieved on August 17, 2021)

[8]The Balinese Alphabet, http://www.babadbali.com/aksarabali/alphabet.htm (Retrieved on August 17, 2021)

"siki kirang siki pateh sareng telas"; 2) In the computer-based transliteration implementation where the symbols available and to ease learning, non-alphanumeric expressions, like the previous example "$1 - 1 = 0$", should be preserved to avoid complexity related to the various phrase expressions that have the same meaning (see the previous aspect); 3) The second aspect was limitedly handled by the supporting computer font, which in this case is the Noto Serif Balinese (NSB) font [9], and needs a special algorithm.

3.1. **The supporting glyphs.** Non-alphanumeric text (including punctuation, mathematical, and general symbols), as the input to the BaliScript web application for the proposed method analysis, used the characters from the desktop/laptop keyboard or the mobile device keypad. Table 1 shows the related Unicode for the analysis, i.e., from the NSB font and mobile keypad. This research covers non-alphanumeric symbols on a mobile device since NSB font uses dedicated Balinese Unicode as a standard Balinese Script font on the computer system including mobile devices. So in the future, the proposed method can be reused on the mobile application as part of the planned ubiquitous learning.

Table 1. The related Unicode for the numerals and non-alphanumeric symbols

NSB Font Unicode																											
Balinese Numeral Unicode	1B50	1B51	1B52	1B53	1B54	1B55	1B56	1B57	1B58	1B59																	
Latin Numeral	0	1	2	3	4	5	6	7	8	9																	
Balinese Script Numeral Glyph	᭐	᭑	᭒	᭓	᭔	᭕	᭖	᭗	᭘	᭙																	
Balinese Non-alphanumeric Unicode	1B5D	1B5E	1B5F																								
Latin Non-alphanumeric	:	,	.																								
Balinese Script Non-alphanumeric Glyph	᭞	᭟	᭠																								
Mobile Keypad Unicode																											
Latin Numeral Unicode	0030	0031	0032	0033	0034	0035	0036	0037	0038	0039																	
Latin Numeral	0	1	2	3	4	5	6	7	8	9																	
Latin Non-alphanumeric Unicode	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F	003A	003B	003C	003D	003E	003F	0040	005B	005C	005D
Latin Non-alphanumeric	!	"	#	$	%	&	'	(	)	*	+	,	−	.	/	:	;	<	=	>	?	@	[	\	]		
Latin Non-alphanumeric Unicode	005E	005F	0060	007B	007C	007D	007E	2260	00A3	00A5	00A9	00AE	00B0	00B6	03C0	00D7	00F7	2018	201C	2022	20AC	2122	0394	221A	2713		
Latin Non-alphanumeric	^	_	`	{	|	}	~	≠	£	¥	©	®	°	¶	π	×	÷	‘	“	•	€	™	Δ	√	✓		

The grey boxes indicate the use of NSB font. On the transliteration, Latin numeral text from the desktop/laptop keyboard or mobile keypad (U+0030-U+0039) must be converted to the Balinese Unicode (U+1B50-U+1B59), appropriately. For the non-alphanumeric symbols, only several punctuations, i.e., ":" (colon, U+003A), "," (comma, U+002C), and "." (period, U+002E) must be converted to the Balinese Unicode (U+1B5D-U+1B5F), appropriately. Other non-alphanumeric symbols should be ignored except for the mathematical symbols.

3.2. **The algorithm.** Algorithm 1 shows the pseudocode for processing of non-alphanumeric text on transliteration to the Balinese Script. For line 2 of the pseudocode, the example is that for the colon symbol ":", its Unicode U+003A must be converted to the Unicode U+1B5D; henceforth through a certain algorithm code, the web application would display it as the Balinese Script "ᬿ". This condition is applied also to line 6 for the mathematical symbol "π" of the non-alphanumeric expression. For line 4 of the pseudocode, the maintained Latin Unicode was displayed only with a numeral in a simple mathematical statement, or in unary operation or binary operation. Other than that, it will be replaced with space " " by using the regular expression matching [19,20]. For line 8 of the pseudocode, the ignoring process for the symbol was conducted by using the validation mechanism to prevent it from entering the data array that was used for the next process.

3.3. **The testing.** Table 2 shows the testing transliteration cases on several non-alphanumeric texts including general symbols, simple mathematical statements, unary operations, and binary operations [21]. Those expressions were constructed by all possible symbols from the desktop/laptop keyboard and mobile keypad based on the related Unicode (Table 1). Arbitrary no space condition means at least there are two characters without space in between. The exception means there is an alphabet or numeral involved.

For example, the alphabet "x" (Unicode U+0078) represents the mathematical multiplication expression since that alphabet is unknown in Balinese Script. That alphabet was used by the desktop/laptop keyboard since there is no non-alphanumeric symbol "×" (Unicode U+00D7), like in the mobile keypad, to represent the mathematical multiplication expression. The testing of the proposed method was conducted on the Model-View-Controller (MVC) -based BaliScript web application [22,23] that was run on Intel hardware platform with Windows Operating System, Apache web server, MySQL database server, and programming language PHP and JavaScript.

Algorithm 1: The non-alphanumeric expression processing

```
1  if the symbol ∈ the punctuation set P = { ":", ",", "."} then
2        change its Latin Unicode to the Balinese Unicode.
3  else if the symbol ∈ the mathematical sign set M1 = {"!", "%", "*", "+", "−",
   "/", "<", "=", ">", "^", "×", "÷", "●", "○", "√", "x"} then
4        maintain its Latin Unicode conditionally.
5  else if the symbol ∈ the mathematical set M2 = {"π"} then
6        change its Latin Unicode to the Balinese Unicode.
7  else then
8        ignore it.
```

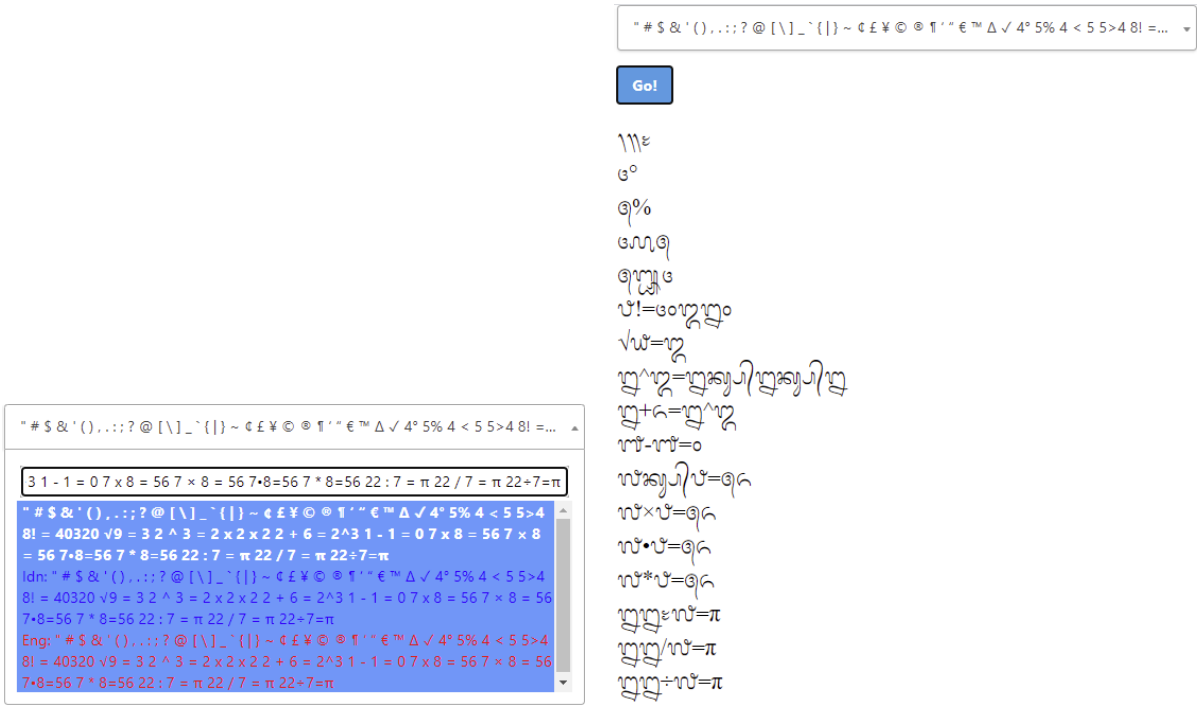
4. **Result and Analysis.** Figure 1 shows the testing result of the proposed method for processing non-alphanumeric text (including punctuation, mathematical, and general symbols) on the learning application of transliteration to the Balinese Script. The testing on the BaliScript transliteration learning web application was based on testing cases in Table 2. Figure 1(a) shows the input of the Latin non-alphanumeric text through the Select input box[9], while Figure 1(b) shows the seventeen lines of the transliteration output (each line has the line break) based on the seventeen lines of the testing input, as shown in Table 2.

Balinese Script writing uses the non-scriptio-continua style [24] traditionally which means the output should fully occupy all of the writing space without paragraph and line break. To ease learning through visual analysis, the non-scriptio-continua style was

[9]Select2 box, https://select2.org (Retrieved on August 17, 2021)

TABLE 2. Testing transliteration cases

No	Case sample	Operation	Remarks
1	" # $ & ' ( ) , . : ; ? @ [¥] _ ` { | } ~ ¢ £ ¥ © ® ¶ ‘ “ € ™ Δ ✓	–	General
2	$4°$	Simple	Degree; No space condition
3	5%	Simple	Percentage; No space condition
4	$4 < 5$	Simple	Less than
5	$5 > 4$	Simple	Bigger than; No space condition
6	$8! = 40320$	Unary	Factorial
7	$\sqrt{9} = 3$	Unary	Square root; No space condition
8	$2\text{^}3 = 2 \times 2 \times 2$	Binary	Square; The exception
9	$2 + 6 = 2\text{^}3$	Binary	Addition; No space condition
10	$1 - 1 = 0$	Binary	Subtraction
11	7 x 8 = 56	Binary	Multiplication; The exception
12	$7 \times 8 = 56$	Binary	Multiplication
13	$7 \bullet 8 = 56$	Binary	Multiplication; No space condition
14	$7 * 8 = 56$	Binary	Multiplication; No space condition
15	$22 : 7 = \pi$	Binary	Division; The exception
16	$22/7 = \pi$	Binary	Division
17	$22 \div 7 = \pi$	Binary	Division; No space condition



(a) (b)

FIGURE 1. Transliteration result of the non-alphanumeric expression: (a) The input through the Select box; (b) the output with the non-scriptio-continua style

used since it is possible by the computer-based approach[10]. Based on the line result, several aspects were analyzed related to the transliteration process that was successfully displayed by the proposed method.

The first line result shows the unconditional display of several punctuation symbols, i.e., ":" (colon, U+003A), "," (comma, U+002C), and "." (period, U+002E). This is in opposite direction with the other non-alphanumeric symbols that were ignored (Algorithm 1) since Balinese Script does not use it by the rule. For the other unconditional display, the mathematical symbol "π" from the mobile non-alphanumeric symbols was also applied, as shown by the fifteenth-seventeenth line result.

The second-third line result shows examples of the simple mathematical expression that was successfully displayed in Balinese Script even though there is no space condition between the numeral and the degree or percentage sign. The algorithm has handled this condition with the result without a difference to the space condition (see the example of the eighth line result that was successfully displayed binary square operation).

The fourth-fifth line result shows examples of the simple mathematical expression that was unsuccessfully displayed the non-alphanumeric expressions in Balinese Script since "<" (less than, U+003C) and ">" (bigger than, U+003E) have already been dedicated by the Balinese font system, each for "᭞" (panten sign, U+1B5A) and "᭟" (pamada sign, U+1B5A) sign. The panten sign was used at the beginning of the letter, story, or verse, while the pamada sign was used at the beginning of religious texts. These cases were considered as one of the limitations for the non-alphanumeric expressions, and also the finding related to this kind of mathematical expression.

The sixth-seventh line result shows examples of the rear and front unary mathematical expression that was successfully displayed in Balinese Script. The rear unary operator of the factorial sign was placed at the rear or the right side after the numeral (operand), while the front unary operator of the square root sign was placed at the front or the left side after the numeral (operand). The transliteration algorithm has taken care of this kind of category of the unary mathematical expression.

The eighth line result shows examples of the binary mathematical expression consist of the left-side and right-side expressions. The left-side expression of square operation was successfully displayed, while the right-side expression was unsuccessfully displayed in the Balinese Script since "x" (alphabet "x", U+0078) has already been dedicated by the Balinese font system, for "ᬓᭀᬲ", i.e., combination of "ᬓ" (syllable "ka", U+1B13), "ᭀ" (adeg-adeg sign, U+1B44), and "ᬲ" (syllable "sa", U+1B32). This case (and also the same case on the eleventh line result) was considered as the other limitation. It is also the finding related to the alphabet "x" that cannot be used for the mathematical expressions transliteration, unlike other signs that can be found on the common desktop/laptop keyboard, i.e., "+", "−", "∗", ":", and "/" (The ninth, tenth, fourteenth, fifteenth, and sixteenth line result, respectively).

The twelfth, thirteenth, and seventeenth line result shows examples of the binary mathematical expression that was successfully displayed in the Balinese Script. These cases show the accommodation of the algorithm for the signs that can be found on the mobile keypad, i.e., "×" (multiplication sign, U+00D7), "•" (multiplication sign, U+2022), and "÷" (division sign, U+00F7). With the support of the non-alphanumerics symbols from the mobile keypad, the proposed method can be easily reused for the transliteration learning mobile application.

5. **Conclusions.** A method for non-alphanumeric text processing on transliteration to the Balinese Script was proposed with the implementation on the web-based learning

[10]CSS white-space, https://www.w3schools.com/cssref/pr_text_white-space.asp (Retrieved on August 17, 2021)

application by using Noto Serif Balinese (NSB) font. The proposed method covers non-alphanumeric symbols on a mobile device since NSB font uses dedicated Balinese Unicode as a standard Balinese Script font on the computer system including mobile devices, which implicates the method can be reused on the mobile application as part of the planned ubiquitous learning. Although there are several limitations because of the font inherent limitation, especially on the mathematical expression, this research added a certain perspective and strengthened the transliteration knowledge, which is part of the local Balinese Language preservation effort in general. Future work is related to improving the proposed method in synergy with the ongoing research on the counterpart alphanumeric text processing on transliteration to the Balinese Script.

Acknowledgment. This work is supported by the Indonesian Ministry of Education, Culture, Research, and Technology through the multi-years Applied Research Grant.

REFERENCES

[1] S. Karimi, F. Scholer and A. Turpin, Machine transliteration survey, *ACM Computing Surveys*, vol.43, no.3, pp.1-46, 2011.

[2] K. Kaur and P. Singh, Review of machine transliteration techniques, *International Journal of Computer Applications*, vol.107, no.20, pp.13-16, 2014.

[3] G. J. Hwang, C. C. Tsai and S. J. H. Yang, Criteria, strategies and research issues of context-aware ubiquitous learning, *Journal of Educational Technology and Society*, vol.11, no.2, 2008.

[4] H. Ogata, Y. Matsuka, M. M. El Bishouty and Y. Yano, LORAMS: Linking physical objects and videos for capturing and sharing learning experiences towards ubiquitous learning, *International Journal of Mobile Learning and Organisation*, vol.3, no.4, pp.337-350, 2009.

[5] G. Indrawan, G. R. Dantes, K. Y. E. Aryanto and I K. Paramarta, Handling of mathematical expression on Latin-to-Balinese script transliteration method on mobile computing, *Proc. of the 5th International Conference on Informatics and Computing (ICIC)*, 2020.

[6] I. G. K. Anom et al., *Balinese – Indonesian Dictionary with Its Latin and Balinese Script (Kamus Bali – Indonesia Beraksara Latin dan Bali)*, Denpasar, Bali Province, 2009.

[7] G. Indrawan, I K. Paramarta, K. Agustini and Sariyasa, Latin-to-Balinese script transliteration method on mobile application: A comparison, *Indonesian Journal of Electrical Engineering and Computer Science*, vol.10, no.3, pp.1331-1342, 2018.

[8] G. Indrawan, I K. Paramarta and K. Agustini, A new method of Latin-to-Balinese script transliteration based on noto sans balinese font and dictionary data structure, *Proc. of the 2nd International Conference on Software Engineering and Information Management (ICSIM)*, 2019.

[9] The Unicode Consortium, *The Unicode Standard Version 13.0 – Core Specification*, 2020.

[10] G. Indrawan, I K. Paramarta, I G. Nurhayata and S. Sariyasa, A method to accommodate backward compatibility on the learning application-based transliteration to the balinese script, *International Journal of Advanced Computer Science and Applications*, vol.12, no.6, pp.280-286, 2021.

[11] G. Indrawan, Sariyasa and I K. Paramarta, A new method of Latin-to-Balinese script transliteration based on Bali Simbar font, *Proc. of the 4th International Conference on Informatics and Computing*, 2019.

[12] P. N. Crisnapati et al., Pasang Aksara Bot: A Balinese Script writing robot using finite state automata transliteration method, *Journal of Physics: Conference Series*, vol.1175, no.1, 2019.

[13] G. Indrawan, K. Setemen, W. Sutaya and I K. Paramarta, Handling of line breaking on Latin-to-Balinese script transliteration web application as part of balinese language ubiquitous learning, *Proc. of the 6th International Conference on Science in Information Technology (ICSITech)*, 2020.

[14] G. Indrawan, I. P. E. Swastika, Sariyasa and I K. Paramarta, 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*, vol.83, pp.7676-7683, 2020.

[15] G. Indrawan, I. G. A. Gunadi, M. S. Gitakarma and I K. Paramarta, Latin-to-Balinese script transliteration: Lessons learned from computer-based implementation, *Proc. of the 4th International Conference on Software Engineering and Information Management (ICSIM)*, 2021.

[16] G. Indrawan, N. N. H. Puspita, I K. Paramarta and Sariyasa, LBtrans-Bot: A Latin-to-Balinese script transliteration robotic system based on Noto Sans Balinese font, *Indonesian Journal of Electrical Engineering and Computer Science*, vol.12, no.3, pp.1247-1256, 2018.

[17] L. H. Loekito, G. Indrawan, Sariyasa and I K. Paramarta, Error analysis of Latin-to-Balinese script transliteration method based on Noto Sans Balinese font, *Proc. of the 3rd International Conference on Innovative Research Across Disciplines (ICIRAD)*, 2020.
[18] I. M. Suatjana, *Bali Simbar Dwijendra*, Yayasan Dwijendra, Denpasar, 2009.
[19] S. Hollos and J. R. Hollos, *Finite Automata and Regular Expressions: Problems and Solutions*, Exstrom Laboratories LLC Co., Longmont, 2013.
[20] L. Groner and G. Manricks, *JavaScript Regular Expressions*, Packt Publishing, Birmingham, 2015.
[21] D. W. Hardy and C. L. Walker, *Applied Algebra: Codes, Ciphers and Discrete Algorithms*, Prentice-Hall, Upper Saddle River, NJ, 2002.
[22] A. Moutaouakkil and S. Mbarki, MVC frameworks modernization approach: Adding MVC concepts to KDM metamodel, *International Journal of Advanced Computer Science and Applications*, 2019.
[23] M. S. Singh, MVC framework: A modern web application development approach and working, *International Research Journal of Engineering and Technology*, 2020.
[24] A. R. Widiarti and R. Pulungan, A method for solving scriptio continua in Javanese manuscript transliteration, *Heliyon*, vol.6, no.4, 2020.