Indonesian Journal of Electrical Engineering and Computer Science
Vol. 12, No. 3, December 2018, pp. 1247~1256
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v12.i3.pp1247-1256

LBtrans-Bot: A Latin-to-Balinese Script Transliteration Robotic System based on Noto Sans Balinese Font

G. Indrawan[1], N. N. H. Puspita[2], I K. Paramarta[3], Sariyasa[4]
[1,2,4]Department of Computer Science, Universitas Pendidikan Ganesha,
Jl. Udayana 11 Singaraja, Bali, Indonesia 81116
[3]Department of Balinese Language Education, Universitas Pendidikan Ganesha, Bali, Indonesia

Article Info

Article history:

Received Feb 21, 2018
Revised Jul 10, 2018
Accepted Aug 29, 2018

Keywords:

Balinese script
Latin script
Noto Sans Balinese
Robotic
Unicode

ABSTRACT

This research aims to preserve Balinese script writing knowledge using technological approach. This across-disciplines research (Computer Science and Balinese Language) contributes on the development of a Latin-to-Balinese script transliteration robotic system that was called LBtrans-Bot. LBtrans-Bot can be used as a learning system to give the transliteration knowledge as one aspect of Balinese script writing. LBtrans-Bot was known as the first system that utilize Noto Sans Balinese font and was developed based on the identified seventeen kinds of special words. LBtrans-Bot consists of the transliterator web application, the transceiver console application, and the robotic arm with its GUI controller application. Through the experiment, LBtrans-Bot has been able to write the 34-pixel font size of the Noto Sans Balinese font from HTML 5 canvas that has been setup with additional 10-pixel length of the width and the height of the Balinese script writing area. Its transliterator gave the accuracy result up to 91% testing cases of The Balinese Alphabet writing rules and examples document. This transliterator result outperformed the best result of the known existing transliterator based on Bali Simbar font, i.e. Transliterasi Aksara Bali, that only has accuracy up to 68% cases of the same testing document. In the future work, LBtrans-Bot could be improved by: 1) Accommodating more complex Balinese script with trade off to the limited writing area of robotic system; 2) Enhancing its transliterator by enriching the database consists of words belong to the seventeen kinds of special words, and implementing semantic relation transliteration.

Corresponding Author:

G. Indrawan,
Department of Computer Science,
Universitas Pendidikan Ganesha,
Jl. Udayana 11 Singaraja, Bali, Indonesia 81116.
Email: gindrawan@undiksha.ac.id

1. INTRODUCTION

There is a concern about the extinction of the Balinese script writing, as one of the Balinese cultural richness [1] since everyday use of that script has largely been replaced by the Latin alphabet. As Stern [2] said that saving the language can be done by cultural, political, and economic approach, another approach was taken, i.e. technological approach by this across-disciplines research (Computer Science and Balinese Language). This research contributes on the development of a Latin-to-Balinese script transliteration robotic system that was called LBtrans-Bot.

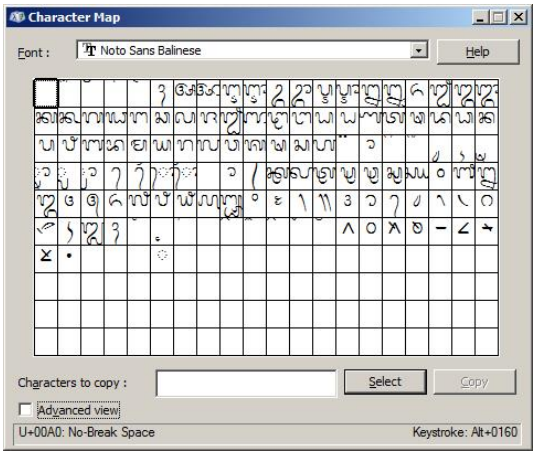
More detail, first contribution is that LBtrans-Bot can be used as a learning system to give the transliteration knowledge [3]-[7] as one aspect of Balinese script writing. Second contribution is that LBtrans-Bot was known as the first system that utilize Noto Sans Balinese font [8] and was developed based

on the identified seventeen kinds of special words. *Dictionary* data structure [9] was used to accomodate those seventeen kinds of special words, as the advance exploration of the thirteen kinds of special words [1]. That data structure was previously used by the authors for fingerprint data discriminator on biometric searching [10], [11]. Third contribution is that its LBtrans-Bot transliterator improves the transliteration accuracy among other existing methods that used Bali Simbar font [12]. Arimbawa et al. [13] has developed its transliteration method on its own Latin-to-Balinese script transliteration robotic system. Sartini et al. [14] has developed its transliteration method on a text-to-digital-image converter desktop application where the output Balinese script was retrieved from pre-collected images that were captured previously from Bali Simbar font display at word processor. Two other methods on Android mobile application [1], each was called Belajar Aksara Bali (Learning Balinese Script), and Transliterasi Aksara Bali (Balinese Script Transliteration).

2. RESEARCH METHOD

2.1. Noto Sans Balinese Font

Noto Sans Balinese font [8] with its Unicode [15] accommodates Balinese Script Complex Behaviours [16] which cover: 1) reordering and splitting; 2) various placement of diacritics; 3) contextual shaping; and 4) complex ligature construction. Figure 1a shows Character Map tool that was used to display Noto Sans Balinese font. By selecting certain character, its related Unicode can be seen at bottom left part of Character Map. Figure 1b shows the Unicode *code point* group allocation 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-adeg*; 7) 1B45–1B4B for additional consonants (Akśara 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.



(a)

Note: At the time of this study, Noto Sans Balinese font is available in version 1.03 with 121 code points and unused 7 reserved code points (grey areas). Its glyphs can be displayed on Microsoft Word document by inserting Symbol, and selecting this font from the Symbol dialog box. This applied if this font has already been installed at the Windows operating system, so it will be listed at the font drop-down list of Symbol dialog box.

(b)

Figure 1. Noto Sans Balinese font: a) the character map; b) the Unicode

2.2. The Architecture

A Latin-to-Balinese script transliteration robotic system based on Noto Sans Balinese font, LBtrans-Bot, consists of the transliterator web application, the transceiver console application, and the robotic arm with its Graphical User Interface (GUI) controller application, as shown by Figure 2. The transliterator was developed based on the identified seventeen kinds of special words, valid Latin vowels, consonants, numerics, and punctuations, conversion of Latin foreign consonants, special vowels, and consecutive vowels. It used the Model-View-Controller architectural pattern. The Model was implemented by using MySQL database, as the repository for the words belong to the seventeen kinds of special words [1]. *Dictionary* data structure was used in the transliterator memory as a place to hold data (words) from the Model. *Dictionary* has *time complexity O*(*1*) regardless of the number of the words save inside this kind of data structure [9]. The View was implemented by using HTML, PHP, CSS, and Bootstrap, mostly for the User Interface responsive design. The Controller was implemented by using JavaScript, mostly for the transliteration

algorithm and as the controller between the Model and the View. The transceiver used batch script and AutoIt script [17] to send data from the transliterator to the GUI controller, which control the Balinese script writing of the robotic arm via USB cable or Bluetooth. The data is actually bitmap (BMP) file consists of Balinese script image that has already been thinned for the technical consideration on writing by the robotic arm. Figure 3 shows eight different patterns for the thinning process and its result on Balinese script image. The robotic arm with its GUI controller used open-source mDrawBot Arduino Robot Building platform [18].

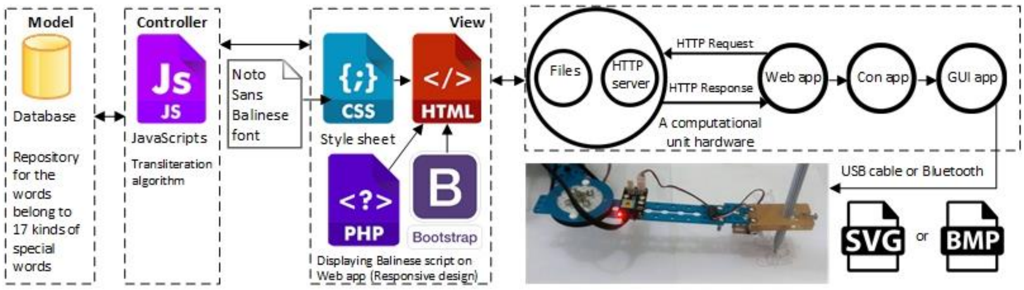


Figure 2. LBtrans-Bot architecture

(a) (b) (c)

Figure 3. Thinning mechanism: a) eight processed different patterns with original pattern (upper) and its related thinned pattern (lower); b) an original image example; c) a thinned image example

2.3. The Algorithm

Figure 4 shows the algorithm pseudocode of a Latin-to-Balinese script transliteration robotic system based on Noto Sans Balinese font, LBtrans-Bot, which actually handles its transliterator web application and its transceiver console application, as shown in Figure 2.

-- LBtrans-Bot transliterator pseudocode --

1. Load **the words** of the seventeen kinds of special words from the repository into the memory (see Figure 2).
   if (a database exists) dictionary from MySQL database
   else dictionary from PHP file
2. Lowercase **the Latin string input**.
3. Filter this **lowercased string** by using Regular Expression (RegExp) to validate used characters.
   validVowels = "aiueoāáǎàâäæìĭīíïîŭūúùûüĕēéèêëœõòŏóōö"
   validConsonants = "bcdfghjklmnpqrstvwxyzßçôṇţşś"
4. Look up the existence of **this validated string** at the *dictionary* (see Figure 2).
   *if (*the validated string *exists)*
   the looked-up string = the special string
   *else if (*the validated string *has the longest common string with* the special string*)*
   the looked-up string = *prefix* of the validated string + the special Latin string + *suffix* of the validated string
   else the looked-up string = the regular string
5. Convert each of the existing foreign consonant(s) of **the looked-up string** to the appropriate valid consonant.
   while ("f", *"v"*, *"q"*, *"x"*, or *"z" exists)*
   convert to *"p"*, *"p"*, *"k"*, *"ks"*, or (*"j"* or *"s"*)
6. Convert each of the existing number(s) of **the converted-foreign-consonant(s) string** to the appropriate Noto Sans Balinese Unicode (for the next reference, it would be written as Unicode) string "&#x1B5*;" where the asterix sign ("*") represents one of the number between *0* to *9* inclusively (see Figure 1b).
7. Convert each of the existing special vowel(s) of **the converted-number(s) string** to to the appropriate Unicode string (see Figure 1b).
   while ("\n", *"\m"*, *"\a"*, *"\aa"*, *"\i"*, *"\ii"*, *"\u"*, *"\ai"*, *"\o"*, *"\au"*, or *"\e" exists)*

convert to the string "ᬀ" (*ardhacandra "n"*), "ᬁ" (*candrabindu "m"*), "ᬅ" (*akara "a"*), "ᬆ" (*akara tedung "ā"*), "ᬇ" (*ikara "i"*), "ᬈ" (*ikara tedung "ī"*), "ᬉ" (*ukara "u"*), "ᬐ" (*aikara "ai"*), "ᬑ" (*okara "o"*), "ᬒ" (*okara tedung "au"*), or "ᬏ" (*ekara "e"*)

8. Convert each of the existing consecutive vowel(s) of **the converted-special-vowel(s) string** to the appropriate consecutive vowel(s).
   while ("ia","iu","ii","ua","uu","ui", or *"ue" exists)*
   convert to *"ya","iyu","iyi","uwa","uwu","uwi",* or *"uwe"*
9. Remove each of the existing space(s) of **the converted-consecutive-vowels string**.
10. Add "h" at the beginning of **the removed-spaces string** that initialized by the vowel to become "h*". The asterix sign ("*") represents one of the vowels, i.e. "a","i","u","e","o", or "æ".
    while (exists as the initial vowel)*
    convert to *"h*"*
11. Convert each of the existing consonant(s) of **the converted-initial-vowel string** to the appropriate Unicode string of that consonant and *adeg-adeg* ("&#x1B**;" + "᭄"). The double asterix sign ("**") represents appropriate last two digits of Unicode of the related consonant (see Figure 1b).
    while ("ng", "ny", "ß", "ç", "nn", "dh", "ph", "th", "tt", "gh", "kh", "ch", "bh", "h", "n", "c", "r", "k", "d", "t", "s", "w", "l", "m", "g", "b", "p", "j", or *"y" exists)*
    convert to the string "&#x1B**;" + "᭄"
12. Convert each of the existing Unicode string of *adeg-adeg* and the following vowel ("᭄*") of **the converted-consonant-adeg-adeg string** to the appropriate Unicode string of sign of that following vowel (see Figure 1b). The asterix sign ("*") represents one of the following vowels, i.e. "ā", "i", "ī", "u", "ū", "e", "ai", "o", "au", "æ", "œ", or "a".
    while ("᭄*" *exists)*
    convert to "ᬵ" (*tedung*), "ᬶ" (*ulu*), "ᬷ" (*ulu sari*), "ᬸ" (*suku*), "ᬹ" (*suku ilut*), "ᬾ" (*taling*), "ᬿ" (*taling repa*), "ᭀ" (*taling tedung*), "ᭁ" (*taling repa tedung*), "ᭂ" (*pepet*), "ᭃ" (*pepet tedung*), or ""
13. Convert each of the remaining vowel(s) of **the converted-adeg-adeg-vowel string** to the appropriate Unicode string of "h" and that remaining vowel ("ᬳ*"). The asterix sign ("*") represents one of the remaining vowel, i.e. "a", "e", "æ", or "o".
    while (exists)*
    convert to "&# x1B33;$"
    where the dollar sign ("$") represents the appropriate Unicode string of that remaining vowel, i.e. "", "ᬾ" (*taling*), "ᭂ" (*pepet*), or "ᭀ" (*taling tedung*).
14. Convert each of the existing Unicode string of illegal combination of "ra" + "æ", "ra" + "œ", "la" + "æ", or "la" + "œ" (both of their regular and appended form) of **the converted-remaining-vowel string** to the appropriate Unicode string of *ra repa* ("ræ" or "rœ") or *la lenga* ("læ" or "lœ"), respectively (see Figure 1b).
    while (exists)*
    convert to $
    where the asterix sign ("*") represents the Unicode string of illegal combination, i.e. "ᬭᭂ" ("ra" + "æ"), "ᬭᭃ" ("ra" + "œ"), "ᬮᭂ" ("la" + "æ"), "ᬮᭃ" ("la" + "œ"), "᭄ᬋ" (*adeg-adeg* + "ræ"), "᭄ᬌ" (*adeg-adeg* + "rœ"), "᭄ᬍ" (*adeg-adeg* + "læ"), or "᭄ᬎ" (*adeg-adeg* + "lœ"), and the dollar sign ("$") represents the appropriate Unicode string of *ra repa* or *la lenga*, i.e. "ᬋ" ("ræ"), "ᬌ" ("rœ"), "ᬍ" ("læ"), "ᬎ" ("lœ"), "ᬺ" ("ræ" sign), "ᬻ" ("rœ" sign), "ᬼ" ("læ" sign), or "&#x1B3D" ("lœ" sign).
15. Convert each of the existing Unicode string of consonants and *adeg-adeg* ("*᭄") of **the converted-illegal-combination string** to the appropriate Unicode string of sound killer $ (see Figure 1b).
    while ("*᭄"*)*
    convert to $
    where the asterix sign ("*") represents the Unicode string of consonants, i.e. "&#x1B02" ("ng"), "ᬃ" ("r"), or "ᬄ" ("h"), while the dollar sign ("$") represents the appropriate Unicode string of sound killer, i.e. "ᬂ" (*cecek*), "ᬃ" (*surang*), or "ᬄ" (*bisah*).
16. Convert punctuations of **the converted-sound-killers string** to the appropriate Unicodes string, i.e. "᭚" (*section*), "᭛" (*honorific section*), "᭝" (*colon*), "᭞" (*comma*), or "&#x1B5F" (*period*).
17. Render **the converted-punctuations string** to the web application display and generate bitmap image of that string for LBtrans-Bot transceiver.

-- LBtrans-Bot transceiver pseudocode --

18. Send the bitmap image to the robotic arm by using batch script that call AutoIt script (Figure 5) to run consecutive click command on GUI controller application. This GUI controller communicate to the robotic arm via USB cable or Bluetooth (Figure 2). The whole operation illustration can be seen at the next Figure 7.

Figure 4. LBtrans-Bot algorithm pseudocode

transceiver.bat

```
@ECHO off
:START
ECHO 1. Checking BMP file for robotic system input...
:BMP_CHECKING
IF NOT EXIST E:\XAMPP7\htdocs\ilkom\lbtransbot\upload\okrobot.bmp GOTO BMP_CHECKING
ECHO 2. Checking PNG file for deletion...
:PNG_CHECKING
IF EXIST E:\XAMPP7\htdocs\ilkom\lbtransbot\upload\okrobot.png GOTO PNG_DELETION
:BMP_PROCESSING
ECHO 4. Sending BMP file to robotic system input...
CD "E:\XAMPP7\htdocs\ilkom\lbtransbot\"
bmp2robot.exe
GOTO BMP_DELETION
:PNG_DELETION
CD "E:\XAMPP7\htdocs\ilkom\lbtransbot\upload\"
DEL *.png /F /Q
ECHO 3. PNG file was deleted...
GOTO BMP_PROCESSING
:BMP_DELETION
CD "E:\XAMPP7\htdocs\ilkom\lbtransbot\upload\"
DEL *.bmp /F /Q
ECHO 5. BMP file was deleted...
GOTO START
```

bmp2robot.au3

```
#include <AutoItConstants.au3>
WinActivate("mDraw")
MouseClick($MOUSE_CLICK_LEFT, 530, 200, 1)
WinWaitActive("Open Svg/Bmp")
MouseClick($MOUSE_CLICK_LEFT, 800, 200, 1)
Send("E:\XAMPP7\htdocs\ilkom\lbtransbot\upload\")
Send("{ENTER}")
Sleep(100)
MouseClick($MOUSE_CLICK_LEFT, 800, 300, 1)
Send("okrobot.bmp")
Send("{ENTER}")
Sleep(100)
MouseClick($MOUSE_CLICK_LEFT, 1200, 630, 1)
Sleep(100)
MouseClick($MOUSE_CLICK_LEFT, 1200, 730, 1)
Sleep(100)
MouseClick($MOUSE_CLICK_LEFT, 700, 200, 1)
```

Figure 5. LBtrans-Bot transceiver console application code: batch script (left); AutoIt script (right)

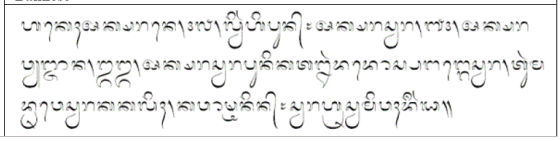
2.4. The Testing

A Latin-to-Balinese script transliteration robotic system based on Noto Sans Balinese font, LBtrans-Bot, was tested as the integrated system consists of the transliterator web application (Table 1 shows its testing schema), the transceiver console application, and the robotic arm with its Graphical User Interface (GUI) controller application as shown in Figure 2. The writing on the robotic arm was tested on simple Balinese script because of the limited writing area belong to the robotic system. This succesful process reflects the succesful of the integrated system mechanism above.

Table 1. Testing cases: a) its category; b) provided sentence and its transliteration

No.	Case	Remark
1st	1-18 19-25	Table 1* basic syllables (no. 1-18) Table 1* examples: word *Bakta* (bring) … *Krama* (member)
2nd	26-34	Table 2* vowel signs examples: word *Kādep* (sold) … *Dwī* (two)
3rd	35-44 45-51	Table 3* independent vowels (no. 1-10) Table 3* examples: word *Akśara* (alphabet)… *Om* (symbol of God)
4th	52-55 56-57	Table 4* illegal combination of syllable - vowel signs (no. 1-4) Table 4* examples: word *Talĕr* (therefore) … *Kĕrĕng* (eat a lot)
5th	58-62	Table 5* semi vowels examples: word *Pak Raman* (Mr. Raman) … *Briag* (laughter)
6th	63-71 72-78	Table 6* *akśara swalalita* (no. 1-9) Table 6* examples: word *Gaṇitri* (chain) … *Laghu* (low tone in singing)
7th	79-83	Table 7* sound killers examples: word *Cengceng* (musical instrument) … *Kapal* (ship)
8th	84-85	Table 8* miscellaneous signs examples: word *Mang* (holy letter) … *Siddham* (perfect)
9th	86-87	Holy symbol *Ongkara* examples: word *Om Swastiastu* (May God blesses you) … *Om Śanti, Śanti, Śanti, Om* (May peace be everywhere)
10th	88-89	Table 9* miscellaneous syllables (no. 1-2)
11th	90-99	Table 10* the digits (digit: 0 - 9)
12th	100-107	Table 11* punctuations: name *Carik* (comma) … Double Quotes
13th	 108-113 114-115 116 117-124	Some variation of usages. Combination of independence vowel *a kara* with vowel signs: vowel *i* … *ŏ* Pairing of *pa kapal* with *suku* or *suku ilut*: syllable *pu* … *phu* Romanization of the inherent sound: word: *Sekala* (real) Usage of *pangangge akśara*: word *Samping* (side)… *Tamblang* (a village's name)
14th	125-146	Table 13* *ligatures* (No. 1-22)
15th	147-150	Abbreviations example: word Bank *Pembangunan Daerah Bali* (Development Bank of Bali Province)… *Ba Pa Da Bali* (BPD Bali)
16th	151	Word boundaries and line break rules

(a)

No	Example
16th	**Latin**
	Akeh akśarane, 47, luir ipun: akśara suara, 14, akśara wianjana, 33, akśara suara punika talĕr dados pangangge suara, tur madrĕwe suara kakalih, kawāśṭanin: suara hrĕswa miwah dīrgha. (Many letters, 47, i.e.: vowels, 14, consonants, 33, those vowels also become vowel signs, and have two type of sounds, each was called: sound *hrĕswa* and *dīrgha*)
	Balinese
	

(b)

Note:
*refer to table number at Sudewa [19]
LBtrans-Bot, was tested as the integrated system consists of the transliterator web application, the transceiver console application, and the robotic arm with its Graphical User Interface (GUI) controller application (Figure 2). All of the application were run on on Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz platform with 8 GB RAM and Windows 7 64-bit Operating System.
The writing on the robotic arm was tested on simple Balinese script because of the limited writing area belong to the robotic system. This succesful process reflects the succesful of the integrated system mechanism above.
At the core of LBtrans-Bot, its transliterator accuracy was tested by using The Balinese Alphabet writing rules and examples document by Sudewa [19], as a project script committee related to the proposal by Eversen and Suatjana [20] for encoding the Balinese script in ISO [21]. Authors' book [22] was also used as a reference. Some of those cases referred to Simpen [23], [24]. Not all of the rules can be tested independently without example (like the appended form of eighteen basic syllables at case 1–18) since provided examples are limited (case 19–25).

At the core of LBtrans-Bot, its transliterator accuracy was tested by using The Balinese Alphabet writing rules and examples document by Sudewa [19] (Table 1a). Authors' book [22] was also used as a reference. Some of those cases referred to Simpen [23], [24]. Not all of the rules can be tested independently without example (like the appended form of eighteen basic syllables at case 1–18) since provided examples are limited (case 19–25). Table 1b shows provided sentence and its transliteration for the testing case 16th (word boundaries and line break rules) at Table 1a.

3. RESULTS AND ANALYSIS

A Latin-to-Balinese script transliteration robotic system based on Noto Sans Balinese font, LBtrans-Bot, has accommodated the seventeen kinds of special words, as the advance exploration of the thirteen kinds of special words [1]. They were handled by LBtrans-Bot transliterator web application where the repository of those words was implemented in the Model (see Figure 2). This seventeen kinds of special words were identified as part of 151 testing cases (were used for accuracy analysis of the transliterator) from The Balinese Alphabet writing rules and examples document by Sudewa [19] (see Table 1). Other than the thirteen kinds of special words from the previous research [1], additional four kinds of special words that were identified on this research, i.e.: 1) The words and their variants (come up due to possibility of different writing of Balinese Letter Ca Laca "cha" using U+1B19, see Figure 1b) that refer to one meaning and should be transliterated the same. For an example: "Chelagi" – "Celagi" (Tamarind fruit); 2) The exception words from the rule about sound killers (*pangangge tengenan*) *cecek* ("ng" using U+1B02, see Figure 1b) or *bisah* ("h" using U+1B04, see Figure 1b) that only appears at the end of a word unless it has the same syllables, e.g. "Cengceng" (musical instrument). For an example: "Angklung" (musical instrument); 3) The words consist of *gantungan* or *gempelan* when a non semi-vowel acts like a semi vowel (the Balinese term is *pluta* that happens very rarely). For an example: "Smerti" (books of Vedha); and 4) The words consist of three consecutive consonants (the term Balinese is *tumpuk telu*) where there is *gantungan* below *gantungan* in their Balinese script. Noto Sans Balinese font does not support that form, so sound killers (*pangangge tengenan*) *adeg-adeg* (using U+1B44, see Figure 1b) can be used even it is not so nice to have it in the middle of a word. For an example: "Tamblang" (a village's name).

Figure 6 shows an example of the second special words that was loaded from LBtrans-Bot's repository/database as shown in Figure 2 to the *dictionary* data structure. All of the various words of this example were transliterated to the same Balinese script by looking up to the core word at the *dictionary* data structure. The lookup process has *time complexity O*(*1*) ignoring of the number of words at that data structure [9].

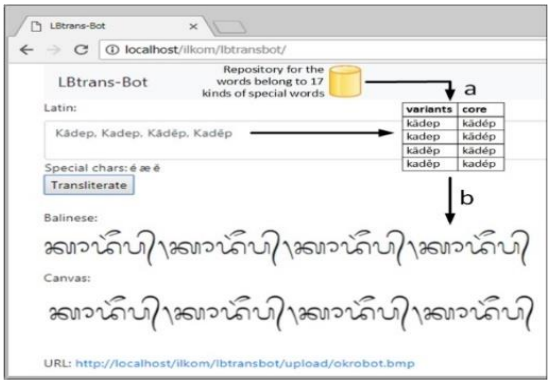


Figure 6. An example of the second kind of special words: a) at the *dictionary* data structure; b) its transliteration result

Figure 7 and Figure 8, each shows an example of LBtrans-Bot process sequence and its drawing. Figure 7 involves several numbered windows (see Figure 2), i.e.:

1. Windows 1 is the LBtrans-Bot transliterator web application that generates bitmap (BMP) file (see Windows 4) of an example of Balinese script (as the result of the transliteration process) and put it at certain folder (see Windows 3). The bitmap file wat setup with 34-pixel size of the Noto Sans Balinese font from HTML 5 canvas with additional 10-pixel length of the width and the height of the Balinese script writing area.

2. Windows 2 is the LBtrans-Bot transceiver console application that detects the generated bitmap file at the certain folder and executes consecutive click command on GUI controller application.
3. Windows 5 is the LBtrans-Bot Graphical User Interface (GUI) controller application communicates to the robotic arm via USB cable or Bluetooth and instructs it to write the Scalable Vector Graphics (SVG) file (see Windows 6) from the bitmap file conversion.

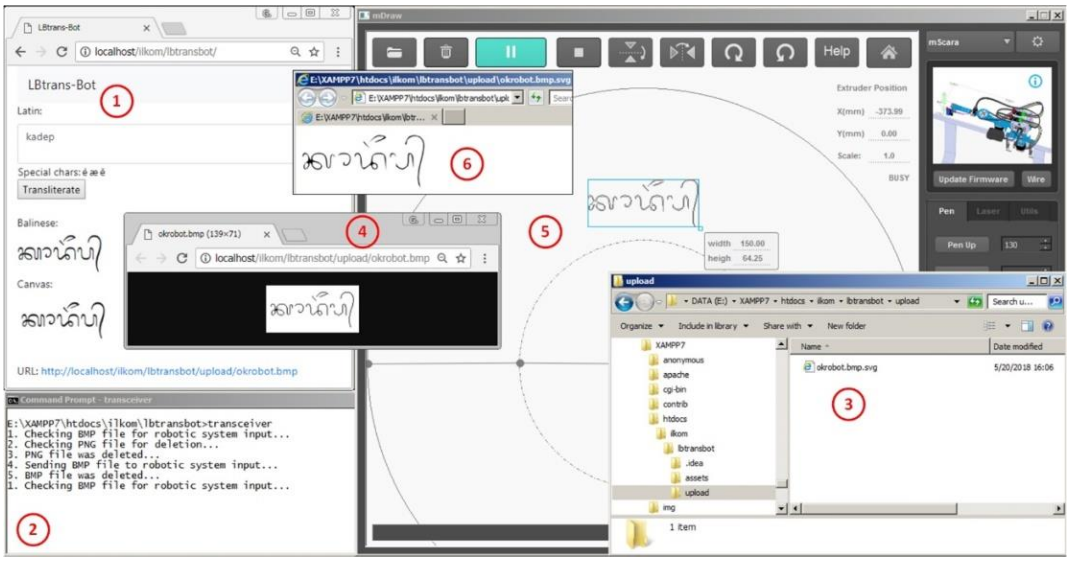


Figure 7. LBtrans-Bot process sequence

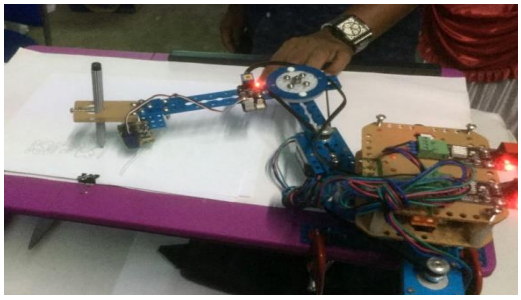


Figure 8. LBtrans-Bot drawing

Table 2 shows the testing result of LBtrans-Bot transliterator, as the core of LBtrans-Bot, where column *Case* represents writing rule or example, as described by Table 1a, and column *Result* shows transliteration result whether correct or incorrect (each was represented by check and cross mark). Based on Table 2, LBtrans-Bot transliterator accuracy has passed over 91% (138 of 151) cases and outperformed Transliterasi Aksara Bali (TAB). Previously, TAB has the best accuracy result up to 68% (103 of 151) cases of the same testing document amongst the existing Latin-to-Balinese script transliteration method [1]. Figure 9 shows LBtrans-Bot transliterator result on the testing case 1st – 6th and 6th – 16th of Table 1a, where testing case numbers were displayed. The next analysis was related to the failed testing cases (Table 1a), why they were happened, and how the possibility to solve them.

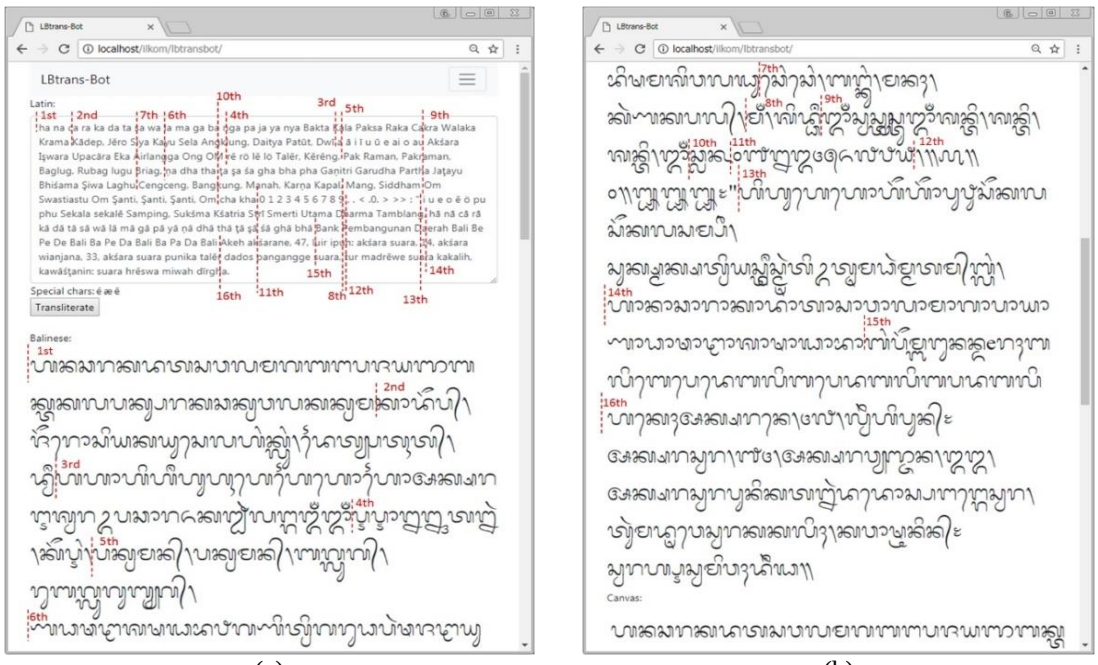
The first kind of failed testing case related to the independent vowel. Ten independent vowels are part of the vowels (*akśara suara*), were used at the initial position of the word (case 35–51 on Table 2, or case 2.35–2.51). Provided examples (case 2.45–2.51) were transliterated correctly. As described previously, not all of the words with a vowel at the initial position use an independent vowel. Each of all of the independent vowels that was written independently (case 2.35–2.44) were transliterated incorrectly as shown in Figure 9a to the reference Table 3a by Sudewa [19]. Table 3b shows the Noto Sans Balinese Unicode for independently written vowels transliteration. At this time, for the sake of algoritma integrity,

LBtrans-Bot transliterator prefered to translate independently written vowels by using Unicode of Consonants and Dependent vowel signs (Table 3b). Future work may address this issue.

The second kind of failed testing case related to the abbreviation. Three different scheme for Balinese abbreviations [19], i.e.: 1) The one endorsed by the government; 2) The one used by I Wayan Simpen A.B. in his schoolbook [24]; and 3) The one less commonly used, but somehow the shortest one. Case 2.147–2.150 consists of phrase "Bank Pembangunan Daerah Bali" (Development Bank of Bali Province) and all of its three abbreviation schemes. Latin abbreviation of that phrase is "BPD Bali" and its three abbreviation schemes, i.e. "Be Pe De Bali", "Ba Pe Da Bali", and "Ba Pa Da Bali" at case 2.148, 2.149, and 2.150, respectively. On all of abbreviations schema there are *cariks* (comma sign) between syllable and/or word that still cannot be accommodated by LBtrans-Bot transliterator (neither do other algorithms) since there is no smart way to differentiate between this abbreviation phrase with non-abbreviation phrase, except all of these abbreviation phrases were manually put into the database as shown in Figure 2, which are huge in number.

Table 2. Testing Results

Case		Result	Case		Result	Case		Result	Case		Result	Case		Result	Case		Result
1th	1	√		27	√		53	√	7th	79	√		105	√		131	√
	2	√		28	√		54	√		80	√		106	√		132	√
	3	√		29	√		55	√		81	√		107	√		133	√
	4	√		30	√		56	√		82	√	13th	108	√		134	√
	5	√		31	√		57	√		83	√		109	√		135	√
	6	√		32	√	5th	58	√	8th	84	√		110	√		136	√
	7	√		33	√		59	√		85	√		111	√		137	√
	8	√		34	√		60	√	9th	86	√		112	√		138	√
	9	√	3rd	35	×		61	√		87	√		113	√		139	√
	10	√		36	×		62	√	10th	88	√		114	√		140	√
	11	√		37	×	6th	63	√		89	√		115	√		141	√
	12	√		38	×		64	√	11th	90	√		116	√		142	√
	13	√		39	×		65	√		91	√		117	√		143	√
	14	√		40	×		66	√		92	√		118	√		144	√
	15	√		41	×		67	√		93	√		119	√		145	√
	16	√		42	×		68	√		94	√		120	√		146	√
	17	√		43	×		69	√		95	√		121	√	15th	147	√
	18	√		44	×		70	√		96	√		122	√		148	×
	19	√		45	√		71	√		97	√		123	√		149	×
	20	√		46	√		72	√		98	√		124	√		150	×
	21	√		47	√		73	√		99	√	14th	125	√	16th	151	√
	22	√		48	√		74	√	12th	100	√		126	√			
	23	√		49	√		75	√		101	√		127	√			
	24	√		50	√		76	√		102	√		128	√			
	25	√		51	√		77	√		103	√		129	√			
2nd	26	√	4th	52	√		78	√		104	√		130	√			



(a) (b)

Figure 9. Testing results: a) the 1st – 6th case; b) the 6th –16th case

Table 3. The Independent Vowels Reference: a) Sudewa; b) Noto Sans Balinese

No	Bali-nese	Name	La-tin	No	Bali-nese	Name	La-tin
1	ᬅ	*a kara*	*a*	6	ᬉ	*u kara tedong*	*ū*
2	ᬆ	*a kara tedong*	*ā*	7	ᬏ	*e kara*	*e*
3	ᬇ	*i kara*	*i*	8	ᬐ	*airsani a*	*ai*
4	ᬈ	*i kara tedong*	*ī*	9	ᬑ	*o kara*	*o*
5	ᬉ	*u kara*	*u*	10	ᬒ	*o kara tedong*	*au*

Note: Actually, it is optional to translate independently written vowels by using independent vowel, or syllable "ha" that was combined with dependent vowel sign.

(a)

Independent vowels

1B05 ᬅ BALINESE LETTER AKARA = a

1B06 ᬆ BALINESE LETTER AKARA TEDUNG = aa ≡ 1B05 ᬅ 1B35 ᭵

1B07 ᬇ BALINESE LETTER IKARA = i

1B08 ᬈ BALINESE LETTER IKARA TEDUNG = ii ≡ 1B07 ᬇ 1B35 ᭵

1B09 ᬉ BALINESE LETTER UKARA = u

1B0A ᬊ BALINESE LETTER UKARA TEDUNG = uu ≡ 1B09 ᬉ 1B35 ᭵

1B0F ᬏ BALINESE LETTER EKARA = e

1B10 ᬐ BALINESE LETTER AIKARA = ai

1B11 ᬑ BALINESE LETTER OKARA = o

1B12 ᬒ BALINESE LETTER OKARA TEDUNG = au ≡ 1B11 ᬑ 1B35 ᭵

Consonants

1B33 ᬳ BALINESE LETTER HA

Dependent vowel signs

1B35 ᭵ BALINESE VOWEL SIGN TEDUNG = aa

1B36 ᭶ BALINESE VOWEL SIGN ULU = i

1B37 ᭷ BALINESE VOWEL SIGN ULU SARI = ii

1B38 ᭸ BALINESE VOWEL SIGN SUKU = u

1B39 ᭹ BALINESE VOWEL SIGN SUKU ILUT = uu

1B3E ᭺ BALINESE VOWEL SIGN TALING = e

1B3F ᭻ BALINESE VOWEL SIGN TALING REPA = ai

1B40 ᭼ BALINESE VOWEL SIGN TALING TEDUNG = o ≡ 1B3E ᭺ 1B35 ᭵

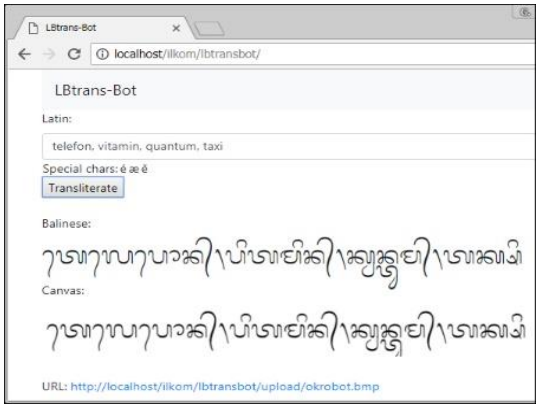
1B41 ᭽ BALINESE VOWEL SIGN TALING REPA TEDUNG = au ≡ 1B3F ᭻ 1B35 ᭵

(b)

On case 2.147, phrase "Bank Pembangunan Daerah Bali" was transliterated correctly. Actually, word "Bank" represents another kind of special words that there is still no research to know the precise list of them [1]. LBtrans-Bot transliterator can handle this kind of special words (see the 5th step of psedocode at The algorithm section) related to foreign sound transliteration, as shown by Figure 10a [25]. Figure 10b shows foreign words transliteration result. At another aspect of the phrase, word "Pembangunan" and "Daerah" is Indonesian words and since no different in writing vowel "e" for different pronunciation "ĕ" (belong to "e" at "Pembangunan") and "e" (belong to "e" at "Daerah"), it is become difficult on transliteration. LBtrans-Bot transliterator can handle this by the same mechanism on the previous special words, like word "Jĕro" (house) [1]. Case 2.149 is the same as case 2.147 where vowel "e" at phrase "Ba Pe Da Bali" is pronounced as "ĕ".

IPA*	Foreign Sound Latin Script	Balinese Language		Example			
					Balinese Language		
		Latin Script	Balinese Script	Foreign Word	Latin Script	Balinese Script	Meaning
[f]	f	p	ᬯ	telefon	telepon	ᬢᬾᬮᬾᬧᭀᬦ᭄	telephone
[v]	v	p	ᬯ	vitamin	pitamin	ᬧᬶᬢᬫᬶᬦ᭄	vitamine
[kw], [k], [q]	q	k	ᬓ	quantum	kuantum	ᬓᬸᬯᬦ᭄ᬢᬸᬫ᭄	quantum
[x]	x	ks	ᬓᬵ	taxi	taksi	ᬢᬓ᭄ᬲᬶ	taxi
[z]	z	j	ᬚ				
[z]	z	s	ᬲ				

(a)



(b)

Figure 10. Foreign sounds transliteration: a) the rules; b) the example results

4. CONCLUSION

A Latin-to-Balinese script transliteration robotic system based on Noto Sans Balinese font, LBtrans-Bot, has been developed and has been able to write the 34-pixel font size of the Noto Sans Balinese font from HTML 5 canvas that has been setup with additional 10-pixel length of the width and the height of the Balinese script writing area. Through the comprehensive accuracy analysis, its transliterator web application gave the accuracy result up to 91% (138 of 151) cases of The Balinese Alphabet writing rules and examples document by Sudewa. This transliterator result outperformed the best result of the known existing transliterator based on Bali Simbar font, i.e. Transliterasi Aksara Bali, that only has accuracy up to 68% (103 of 151) cases of the same testing document. In the future work, LBtrans-Bot could be improved by: 1) accommodating more complex Balinese script with trade off to the limited writing area of robotic system; 2)

enhancing its transliterator by enriching the database consists of words belong to the seventeen kinds of special words by using identified those words from the existing Balinese script dictionary from Bali Province government, and implementing semantic relation transliteration.

ACKNOWLEDGEMENTS

This research was supported by Indonesian Ministry of Research, Technology, and High Education through Grant No. 113/UN48.15/LT/2018. The transliterator web application module of this LBtrans-Bot has already been hosted at server of Computer Science Graduate Program of Universitas Pendidikan Ganesha as a JavaScript-based web application for the purpose of learning and testing for the further improvement.

REFERENCES

[1] 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.
[2] I. N. Jampel, G. Indrawan, and I. W. Widiana, "Accuracy Analysis of Latin-to-Balinese Script Transliteration Method," *International Journal of Electrical and Computer Engineering*, vol. 8, no. 3, 2018.
[3] S. Karimi, F. Scholer, and A. Turpin, "Machine transliteration survey," *ACM Computing Surveys*, 2011.
[4] N. S. Kharusi and A. Salman, "The English Transliteration of Place Names in Oman," *Journal of Academic and Applied Studies*, vol. 1, no. 3, pp. 1–27, 2011.
[5] Y. Marton and I. Zitouni, "Transliteration normalization for Information Extraction and Machine Translation," *Journal of King Saud University - Computer and Information Sciences*, 2014.
[6] K. Kaur, "Review of Machine Transliteration Techniques," *International Journal of Computer Applications*, vol. 107, no. 20, 2014.
[7] M. S. H. Ameur, F. Meziane, and A. Guessoum, "Arabic Machine Transliteration using an Attention-based Encoder-decoder Model," in *Procedia Computer Science*, 2017.
[8] Google, "Google Noto Fonts," 2014. [Online]. Available: https://www.google.com/get/noto/#sans-bali. [Accessed: 10-Aug-2018].
[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Boston: Massachusetts Institute of Technology, 2009.
[10] G. Indrawan, S. Akbar, and B. Sitohang, "Parallel Processing for Fingerprint Feature Extraction," in *International Conference on Electrical Engineering and Informatics (ICEEI)*, 2011.
[11] G. Indrawan, S. Akbar, and B. Sitohang, "On Analyzing of Fingerprint Direct-Access Strategies," in *International Conference on Data and Software Engineering (ICoDSE)*, 2014.
[12] Made Suatjana, "Bali Simbar," 1999. [Online]. Available: http://www.babadbali.com/aksarabali/ balisimbar.htm. [Accessed: 10-Aug-2018].
[13] I. G. N. P. Arimbawa, I. N. Sukajaya, and G. Indrawan, "Pengembangan Robot Penulis Aksara Bali Berbasis Lego Mindstrom NXT [The Development of Balinese Script Writing Robot based on Lego Mindstrom NXT]," in *Seminar Nasional Riset Inovatif (Senari)*, 2017.
[14] I. D. A. M. Sartini, M. W. A. Kesiman, and I. G. M. Darmawiguna, "Pengembangan Text to Digital Image Converter Untuk Dokumen Aksara Bali [The Development of Text to Digital Image Converter for Balinese Script Document]," in *Seminar Nasional Pendidikan Teknik Informartika (Senapati)*, 2012, pp. 280–296.
[15] The Unicode Consortium, *The Unicode Standard, Version 5.0*. Addison-Wesley Professional, 2006.
[16] B. B. Narendra, "Pembuatan Smart Font Aksara Bali [The Development of Balinese Script Smart Font], Undergraduate Thesis," Institut Teknologi Bandung, Bandung, 2008.
[17] AutoIt, "AutoIt Scripting Language," 2018. [Online]. Available: https://www.autoitscript.com/site/ autoit/. [Accessed: 10-Aug-2018].
[18] Makeblock, "mDrawBot: A Four-in-One Drawing Robotic Kit," 2013. [Online]. Available: https://github.com/Makeblock-official/mDrawBot. [Accessed: 10-Aug-2018].
[19] Ida Bagus Adi Sudewa, "The Balinese Alphabet," 2003. [Online]. Available: http://www.babadbali.com/aksarabali/alphabet.htm. [Accessed: 10-Aug-2018].
[20] M. Eversen and I. M. Suatjana, "N2908: Proposal for Encoding the Balinese Script in the UCS." 2005.
[21] ISO, "ISO/IEC International Standard ISO/IEC 10646: Information Technology - Universal Coded Character Set (UCS)." Switzerland, 2010.
[22] G. Indrawan and I. K. Paramarta, *Komputerisasi Transliterasi Teks Latin ke Aksara Bali [Computerized Transliteration of Latin Text to Balinese Script]*. Jakarta: Raja Grafindo, 2017.
[23] I. W. Simpen, *Pasang Aksara Bali [Balinese Script Construction]*. Denpasar: Upada Sastra, 1995.
[24] I. W. Simpen, *Purwa Aksara [Script for the Beginners]*. Denpasar: Upada Sastra, 2002.
[25] I Nengah Tinggen, *Celah-Celah Kunci Pasang Aksara Bali [Key Techniques for Balinese Script Contruction]*. Singaraja: Rhika Dewata, 1994.