ICIC Express Letters
Volume **17**, Number **7**, July **2023**

ICIC International ©2023 ISSN 1881-803X
pp. **725–733**

DERIVATIVE WORD CONVERSION METHOD TO BALINESE SCRIPT ON MOBILE COMPUTING

Gede Indrawan[1,*], Cokorda Oka Birawidya[1], Luh Joni Erawati Dewi[1]
Ketut Agustini[1], I Gede Aris Gunadi[2] and I Ketut Paramarta[3]

[1]Engineering and Vocational Faculty
[2]Faculty of Mathematics and Natural Sciences
[3]Faculty of Languages and Arts
Universitas Pendidikan Ganesha
Jalan Udayana 11, Singaraja, Bali 81116, Indonesia
cokordabirawidya@gmail.com
{ joni.erawati; ketutagustini; igedearisgunadi; ketut.paramarta }@undiksha.ac.id
*Corresponding author: gindrawan@undiksha.ac.id

Received August 2022; accepted November 2022

Abstract. *Conversion of derivative words to Balinese Script has never been studied yet. This multi-discipline collaboration research on that topic is part of the knowledge preservation efforts of the local Balinese language in Indonesia through technology. The proposed method for that kind of conversion, as the main contribution of this work, was based on two existing rules, i.e., 1) a base word of Latin text is converted using standard or special rules to a base word of Balinese Script; and 2) a derivative word related to the previous base word adheres with the first condition. This research was done on a mobile conversion application as a research product for ubiquitous learning. The Balinese Script outputs used a computer font with standard Unicode to display the script. Testing on the proposed method provided conversion results as expected on the derived words testing dataset that represents usual patterns related to the addition of prefix and/or suffix to the base word.*

Keywords: Derivative word, Conversion, Balinese Script, Mobile computing, Unicode

1. **Introduction.** As part of this research area [1,2], conversion knowledge to Balinese Script needs to be preserved. It is part of the endangered local Balinese language in Indonesia with its preservation regulation [3]. This research joined the effort through the technological approach, by proposing a method for derivative word conversion to Balinese Script. Understanding comprehensively the conversion handling of this kind of input text is the objective and the main contribution of this work. The main contribution includes the academic contribution on applied finite state automata [4,5] (related to conversion to a base word of Balinese Script) combined with a rule-based algorithm (related to conversion of derivative word of Balinese Script). Moreover, the proposed method was implemented on mobile computing through a mobile conversion application as a research product and is important for ubiquitous learning [6,7] of Balinese language education. This research has the urgency to succeed in the national research program, where this research is included in one of the ten research focus areas (Information and Communication Technology, ICT), research themes (improving ICT content), and priority research topics (information data on various forms of local wisdom in Indonesia) [8]. Compatible Balinese Unicode font [9-11] was used by the mobile application to display the Balinese Script as the conversion result.

Several works related to this researches on conversion to Balinese Script [12-22]. Balinese fonts, i.e., Bali Simbar (BS) [23], BS Dwijendra (BSD) [24], Noto Sans Balinese

DOI: 10.24507/icicel.17.07.725

(NSBa), and NSB [10], were used to display the Balinese Script output. Research in [13] with dataset referring to [25] used BS font. The development of a writing robotic system [14] and handling of conversion line breaking [15] also used this font. [16] used BSD font with the same dataset in [13] and [26]. The mathematical expression conversion [17] and ten learned lessons on computer-based transliteration [18] also used this font. NSBa font was used in [19] with the same dataset in [13]. Other development of a writing robotic system [20] and other conversion algorithm analysis [21] also used this font. Studies of conversion compatibility with the standard rule applied [12,26] and non-alphanumeric conversion [22] were conducted by using NSB font.

None of those related works exposed comprehensively related to derivative word conversion to Balinese Script. This becomes the motivation of this work with its several contributions, as described previously. The challenge faced in this study is shown in Figure 1. It shows examples of base words with their related derivative words, both in Latin text and Balinese Script, inside dashed-line rectangles from a sample ground truth dataset [26]. The problem lies in representing that dataset in the form of digital data (refer to the database, for the next discussion) efficiently for the conversion process. It is more efficient to have derivative word conversion to Balinese Script if the database only contains its related base word. Figure 1 shows on a small scale (a page from the overall 829-page dataset) how the acquisition of all derivative words to the database will simply occupy additional 27 records of their related 29 records of base words (additional overhead of almost 100%). As a note, base words marked with "I" and "II" in Figure 1 need separate records to accommodate different conversion and/or translation results for the language phenomenon homonym, homophone, and homograph [27-29]. As not shown in Figure 1, words with the same writing of "e" but have different sounds [ə] or [e] [30] also were accommodated in [26] related to homograph words.

Several sections of this paper comprise Introduction, Method, Result and Discussion, and Conclusions. Section Introduction describes the problem, related works, and motivation in this work; Section Method covers the challenge, the algorithm, and the testing; Section Result and Discussion provides the testing result and related discussion; Section 4 (Conclusions) consists of some important conclusion points.

2. **Method.**

This section describes 1) the proposed algorithm for the solution to the challenge faced in this study of derivative word conversion to Balinese Script (described previously); and 2) the testing process.

2.1. **The algorithm.**

The algorithm takes care of two existing rules. First, a base word of Latin text is converted using standard or special rules to a base word of Balinese Script. Second, a derivative word related to its previous base word also has the same conversion condition as the first rule.

Those two rules are existing rules in conversion to Balinese Script. The novelty of this work lies in how the method was developed for processing those two existing rules which have not been studied yet.

The algorithm involves displaying the NSB font for the Balinese Script conversion result. For the first rule example (see Figure 1), the base word "wasta" (noun: name) [26] should have a special conversion "ᬯᭂ", rather than a default conversion "ᬯᬲ᭄ᬢ". The combination of consonant and vowel "sta" should be converted to "ᭂ" using Unicode of syllable Sa Sapa (U+1B31) and Ta Tawa (U+1B23) [11]. It should not be converted to "᬴᭄" using Unicode of syllable Sa (U+1B32) and Ta (U+1B22). For the second rule example (also in Figure 1), the related derivative word "kawastanin" (named) [26] should also have a special conversion "ᬓᬯᭂᬦᬶᬦ", rather than a default conversion "ᬓᬯᬲ᭄ᬢᬦᬶᬦ". Sylablle

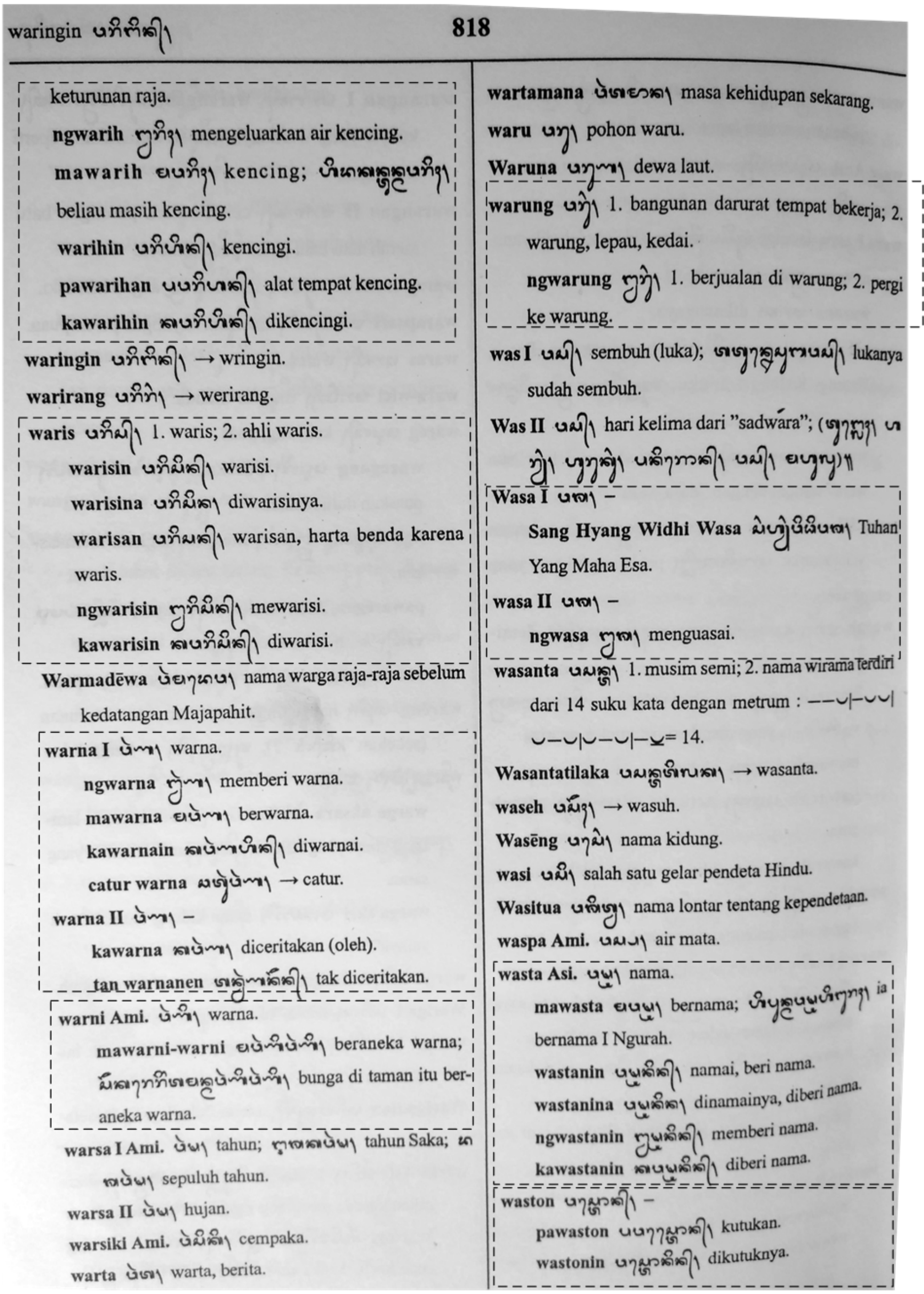

waringin 818
keturunan raja.
ngwarih mengeluarkan air kencing.
mawarih kencing; beliau masih kencing.
warihin kencingi.
pawarihan alat tempat kencing.
kawarihin dikencingi.
waringin → wringin.
warirang → werirang.
waris 1. waris; 2. ahli waris.
warisin warisi.
warisina diwarisinya.
warisan warisan, harta benda karena waris.
ngwarisin mewarisi.
kawarisin diwarisi.
Warmadēwa nama warga raja-raja sebelum kedatangan Majapahit.
warna I warna.
ngwarna memberi warna.
mawarna berwarna.
kawarnain diwarnai.
catur warna → catur.
warna II –
kawarna diceritakan (oleh).
tan warnanen tak diceritakan.
warni Ami. warna.
mawarni-warni beraneka warna; bunga di taman itu beraneka warna.
warsa I Ami. tahun; tahun Saka; sepuluh tahun.
warsa II hujan.
warsiki Ami. cempaka.
warta warta, berita.
wartamana masa kehidupan sekarang.
waru pohon waru.
Waruna dewa laut.
warung 1. bangunan darurat tempat bekerja; 2. warung, lepau, kedai.
ngwarung 1. berjualan di warung; 2. pergi ke warung.
was I sembuh (luka); lukanya sudah sembuh.
Was II hari kelima dari "sadwara";
Wasa I –
Sang Hyang Widhi Wasa Tuhan Yang Maha Esa.
wasa II –
ngwasa menguasai.
wasanta 1. musim semi; 2. nama wirama terdiri dari 14 suku kata dengan metrum : = 14.
Wasantatilaka → wasanta.
waseh → wasuh.
Wasēng nama kidung.
wasi salah satu gelar pendeta Hindu.
Wasitua nama lontar tentang kependetaan.
waspa Ami. air mata.
wasta Asi. nama.
mawasta bernama; ia bernama I Ngurah.
wastanin namai, beri nama.
wastanina dinamainya, diberi nama.
ngwastanin memberi nama.
kawastanin diberi nama.
waston –
pawaston kutukan.
wastonin dikutuknya.

Figure 1. Examples of base words with their related derivative words (inside dashed-line rectangles) from a sample ground truth dataset

“ᬓ” (U+1B13) forms the prefix “ka”, while sylablle “ᬦ” (U+1B26), vowel sign *ulu* “◌ᬶ” (U+1B44), “ᬦ”, and sound killer *adeg-adeg* “◌᭄” (U+1B3E) form the suffix “nin”.

Figure 2 shows the algorithm that generates SQL instruction to the database for the example of variable *word_text* = “kawastanin” with its variable *word_text_length* of 10 (future discussion would directly use the italic variable name for short). The first step sets *min_length* equal to 3, as a minimum number of characters of a base word to be searched in the database. This step also sets a flag of *first* equal to *true* (to generate SQL instruction of UNION ALL SELECT from lines 5-11, see the next Figure 3).

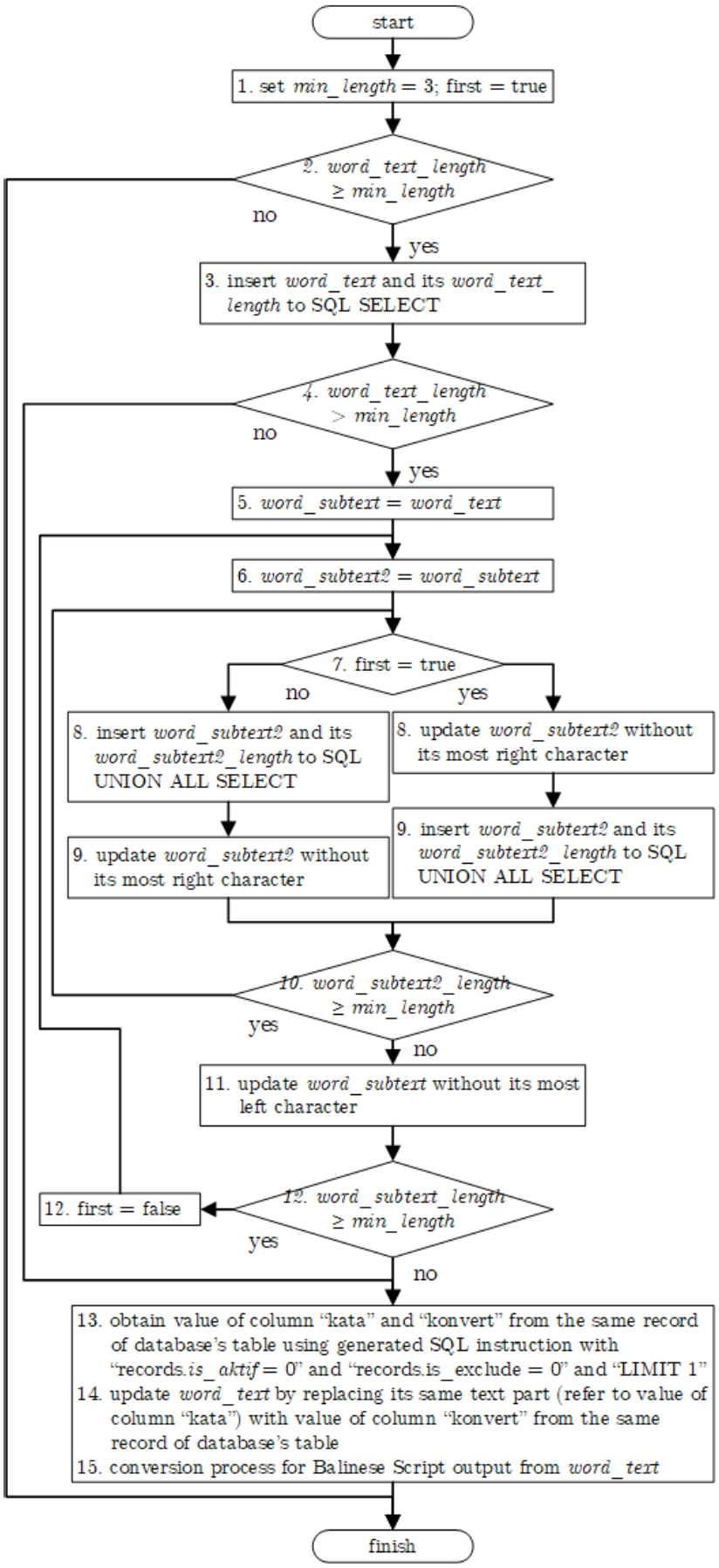


FIGURE 2. The algorithm that generates an SQL instruction to the database

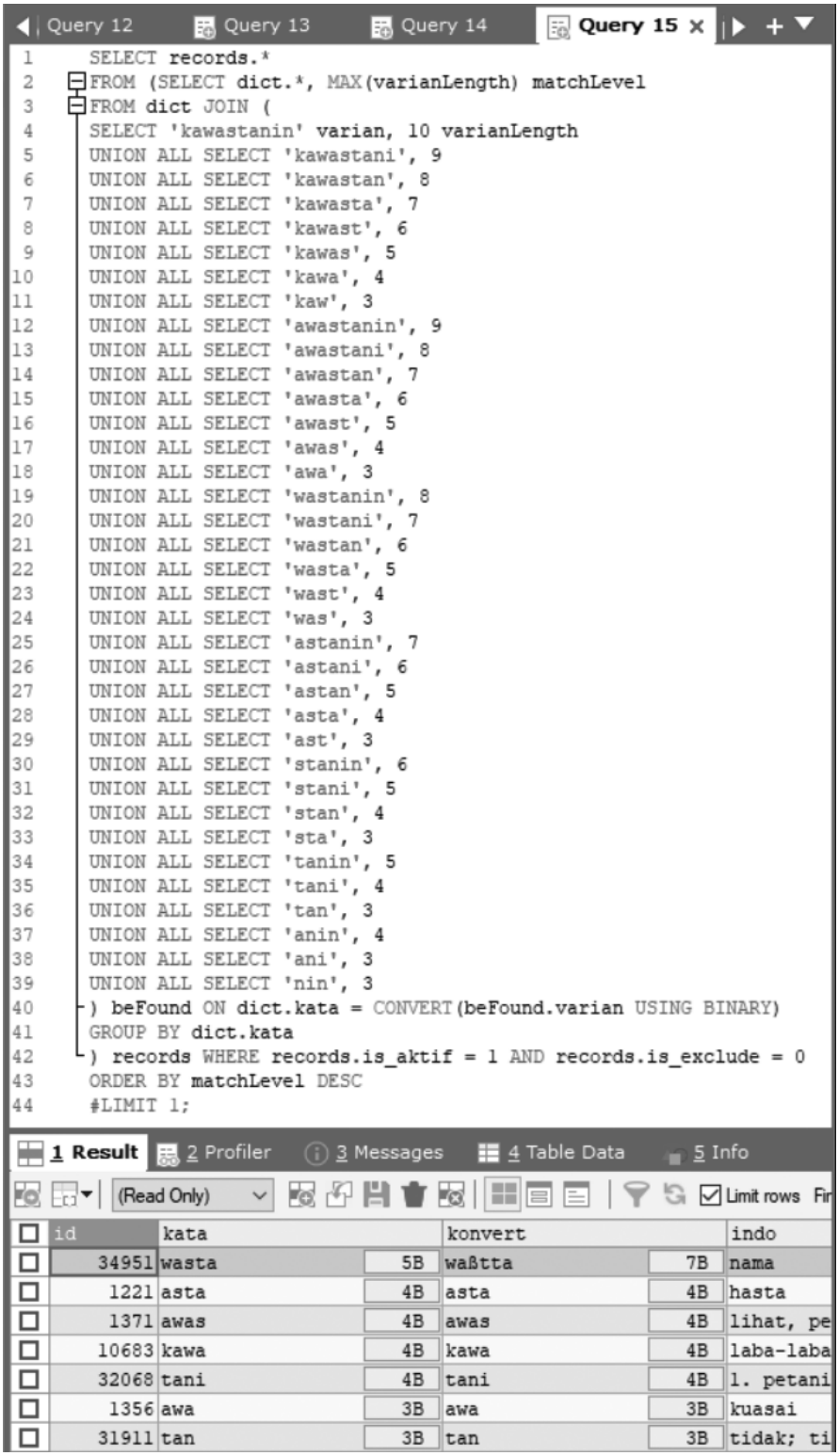


FIGURE 3. An example of algorithm-generated SQL instruction to the database

The second step checks whether *word_text_length* is bigger than or equal to *min_length*. If *NO*, the algorithm is ended. If *YES*, the third step inserts *word_text* and its *word_text_length* to SQL instruction, i.e., "SELECT 'kawastanin' varian, 10 varianLength". The fourth step checks whether *word_text_length* is bigger than *min_length*. If *NO*, the algorithm goes to the thirteenth step forward (described further). If *YES*, the next fifth and sixth steps set *word_subtext* equal to *word_text*, i.e., "kawastanin" and *word_subtext2* equal to *word_subtext*, i.e., "kawastanin", respectively. The seventh step checks whether

the flag "first" is equal to *true*. If *NO*, the next eighth and ninth steps insert *word_subtext2* and its number of characters to SQL instruction (see line 5 of Figure 3); and update *word_subtext2* = "kawastan" with *word_subtext2_length* = 8, respectively. If *YES*, the ninth step of the *NO* path was executed first then followed by the eighth step of the *NO* path. The tenth step checks whether *word_subtext2_length* (for *word_subtext2* = "kawastan") is bigger than *min_length*. If *YES*, the algorithm does another round of iteration starting at the seventh step. If *NO*, the eleventh step updates *word_subtext* = "awastanin" with *word_subtext_length* = 9. The twelfth step checks whether *word_subtext_length* (for *word_subtext* = "awastanin") is bigger than and equal to *min_length*. If *YES*, the algorithm sets the flag "first" equal to *true* and does another round of iteration starting at the sixth step. If *NO*, the thirteenth step obtains the value of column "kata" and column "konvert" (known as Balinese code) from the same record of the database's table using generated SQL instruction with "records.*is_aktif* = 0" and "records.is_exclude = 0" and "LIMIT 1".

Figure 3 shows the complete variants of *word_subtext2* that were searched within column "kata" of records in the database. For the purpose of analysis, the retrieved result of the algorithm-generated SQL instruction contains several records since line 15 was commented. On the proposed method, just one record was retrieved that has the highest match level for the searched base word.

Finally, the fourteenth step updates *word_text* = "kawaßttanin" that was fed into the fifteenth step with the conversion result of "ᬓᬯᬲ᭄ᬢᬦᬶᬦ᭄". The additional contribution related to the translation feature uses the value of column "indo" and column "english" (not shown in Figure 3).

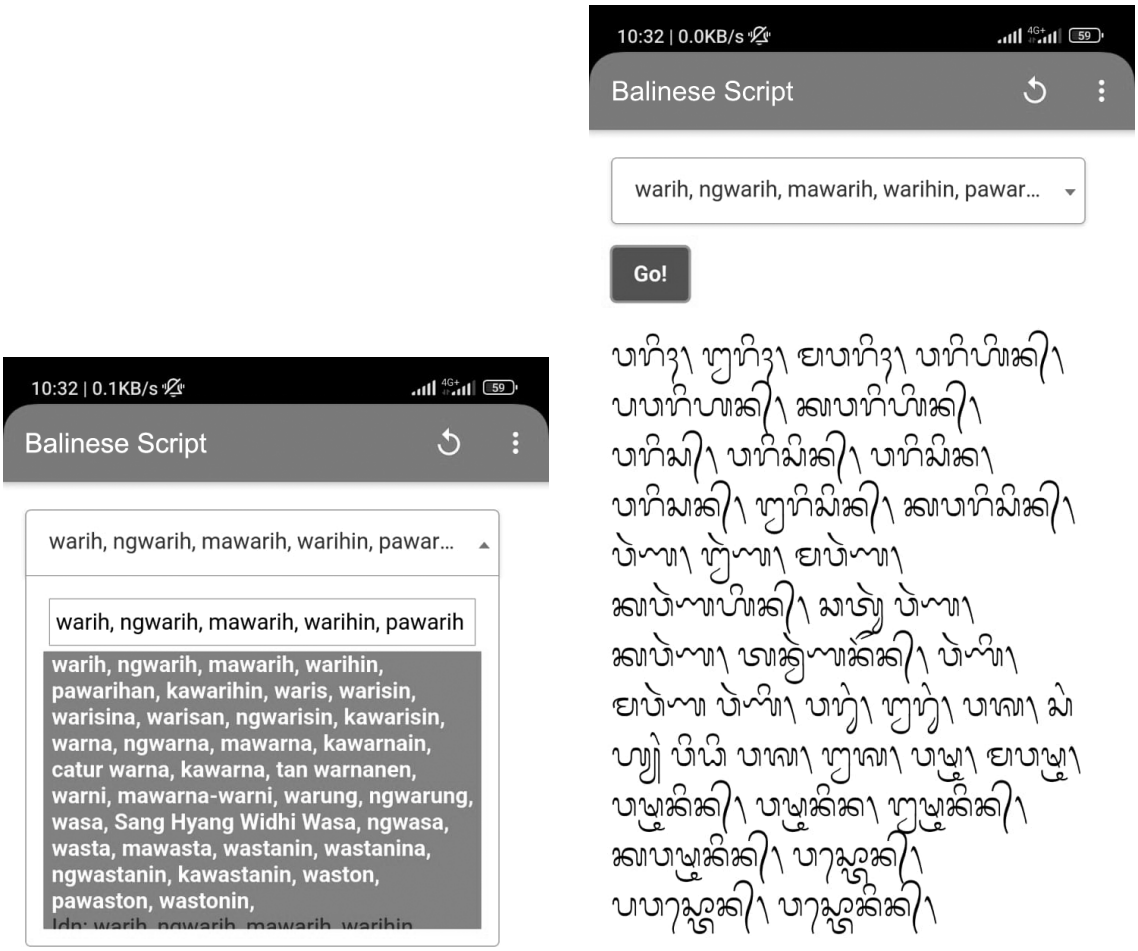
2.2. **The testing.** The testing was held on the mobile application installed on an 8 GB RAM mobile device with a hardware platform of 8 cores CPU (2 ARM Cortex-A76 @2.05 GHz and 6 ARM Cortex-A55 @2.05 GHz) and 64-bit Android 12 OS. In functionality, that mobile application (client side) was developed with Flutter and connected to the web application (server side) that was developed with PHP combined with JavaScript. The server side uses several software platforms, including Apache 2 (Linux64) for the web server and MySQL 10.5 for the database server.

A small dataset shown by the previous Figure 1 was used for the testing conversion cases that represent most types of derivative words consisting of a prefix and/or suffix in the Balinese language.

3. **Result and Discussion.** The proposed method was tested on the conversion mobile application, as shown in Figure 4. Several concerns were discussed on the Balinese Script result that was successfully displayed using dedicated Balinese Unicode font.

The derivative word "ngwastanin" (verb: name), see Figure 1, was formed by a prefix "ng" (which constructs a semivowel "ngw" with the first letter of the base word [26,31]) and a suffix "in". The proposed method successfully converted it to "ᬗ᭄ᬯᬲ᭄ᬢᬦᬶᬦ᭄" based on the correct code "ngwaßttanin" where the code "waßtta" (see Figure 2) came from base "wasta" ("ᬯᬲ᭄ᬢ"). In Balinese Script "ᬗ᭄ᬯᬲ᭄ᬢᬦᬶᬦ᭄", the syllable "wa" (Balinese Letter Wa, U+1B2F) was converted into its appended form "ᭀ" (because of following consonant cluster "ng" [26]) rather than using its regular form "ᬯ".

Other derivative words consisting of semivowel have the same conversion process as the previous word "ngwastanin", including the word "ngwarisin" (verb: inherit), "ngwarih" (verb: pee), "ngwarna" (verb: color), "ngwarung" (verb: (1) sell something in a little shop; (2) go buy something in a little shop), and "ngwasa" (verb: control). The first derivative word has a prefix and suffix, and the rest have a prefix only.



(a) (b)

FIGURE 4. Conversion process: (a) Input; (b) the Balinese Script result displayed using dedicated Balinese Unicode font

The derivative word "kawastanin" (see the previous algorithm section) was converted successfully to "ᬓᬯᬲ᭄ᬢᬦᬶᬦ᭄". A certain aspect should be taken care of related to the existence of other base words that were retrieved by using SQL instruction "UNION ALL SELECT" before the correct base word, as shown by lines 5-21 of Figure 3. The unexpected existing base word "kawas" (noun: a kind of offering) was retrieved first (line 9) rather than the expected base word "wasta" (line 22) (This condition was not shown in Figure 3 since "records.is_aktif" of "kawas" was not set to "1", see line 42). In the case of unexpected base words found, the derivative word "kawastanin" should be registered in the database so the algorithm can retrieve the correct Balinese code "kawaßttanin" in the first place by using "SELECT" (see line 4 of Figure 3). That Balinese code is then used in the fifteenth step of the algorithm for the conversion process (see Figure 2).

Another exploration of derivative words is related to the kind of derivational suffix "in" and "ina" that makes a derivative word differ in word type related to the base word. The suffixes "in" and "ina" are counterpart suffixes where the first one makes a base word as an active verb while the other makes a base word as a passive verb. From the dataset, these derivative words can be converted as expected and include the word "warisin" (verb: inherit), "warisina" (verb: inherited), "wastanin" (verb: name), and "wastanina" (verb: named).

Phrases with or without a derivative word, like "catur warna" (noun: four colors) and "Sang Hyang Widhi Wasa" (God Almighty), were processed on an iteration based on the number of words construct that phrases.

Based on several previous concerns, other derivative words for the conversion process have the same patterns related to the addition of prefix and/or suffix to the base word, and should be converted as expected by the proposed method.

4. **Conclusions.** Several minor concerns were exposed during the testing of the algorithm. In general, the expected result was given on the derived words testing dataset that represents usual patterns related to the addition of a prefix and a suffix of the base word. The evaluation of the proposed method compared to those in the literature should give a more accurate result since the proposed method accommodates the derived word conversion that has never been studied yet before. More detailed evaluation can be done in the future work along with enhancing this method related to loan word handling that needs to be done since the language in nature is a dynamic field because new words appear and need to be converted correctly. Derived words in the opposite direction of conversion (from Balinese Script to Latin text) also become a concern based on lessons learned from this work.

Acknowledgment. The authors gratefully acknowledge the support of the Indonesian Ministry of Education, Culture, Research, and Technology for the research funding in the area of technology for information data on various forms of local wisdom.

REFERENCES

[1] S. Karimi, F. Scholer and A. Turpin, Machine transliteration survey, *ACM Comput. Surv.*, vol.43, no.3, pp.1-46, 2011.
[2] K. Kaur and P. Singh, Review of machine transliteration techniques, *Int. J. Comput. Appl.*, vol.107, no.20, pp.13-16, 2014.
[3] Bali Government, *Bali Governor Regulation No. 80 on Protection and Usage of Balinese Language, Script, and Literature*, 2018, https://jdih.baliprov.go.id/produk-hukum/peraturan/abstrak/24665, Accessed on Aug. 01, 2022.
[4] J. Wang, *Formal Methods in Computer Science*, CRC Press, Boca Raton, 2019.
[5] S. Hollos and J. R. Hollos, *Finite Automata and Regular Expressions: Problems and Solutions*, Exstrom Laboratories LLC, Longmont, CO, 2013.
[6] G. J. Hwang, C. C. Tsai and S. J. H. Yang, Criteria, strategies and research issues of context-aware ubiquitous learning, *J. Educ. Technol. Soc.*, vol.11, no.2, 2008.
[7] 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, *Int. J. Mob. Learn. Organ.*, vol.3, no.4, pp.337-350, 2009.
[8] Kemendikbudristek, *Guideline of Research and Community Service*, Edition XIII Revision, Jakarta, 2021.
[9] Google, *Google Noto Serif Balinese*, 2021, https://github.com/googlefonts/noto-fonts/blob/master/unhinted/ttf/NotoSerifBalinese/NotoSerifBalinese-Regular.ttf, Accessed on Aug. 01, 2022.
[10] The Unicode Consortium, *The Unicode Standard Version 13.0 – Core Specification*, CA, 2020.
[11] *The Unicode Consortium, Balinese Unicode Table*, 2020, http://unicode.org/charts/PDF/U1B00.pdf, Accessed on Aug. 01, 2022.
[12] G. Indrawan, I. K. Paramarta, I. G. Nurhayata and Sariyasa, A method to accommodate backward compatibility on the learning application-based transliteration to the Balinese Script, *Int. J. Adv. Comput. Sci. Appl.*, vol.12, no.6, pp.280-286, 2021.
[13] G. Indrawan, Sariyasa and I. K. Paramarta, A new method of Latin-to-Balinese Script transliteration based on Bali Simbar font, *Proc. of International Conference on Informatics and Computing*, 2019.
[14] P. N. Crisnapati et al., Pasang aksara Bot: A Balinese Script writing robot using finite state automata transliteration method, *J. Phys. Conf. Ser.*, vol.1175, no.1, 2019.
[15] 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 International Conference on Science in Information Technology*, 2020.
[16] 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, *Proc. of the International Conference on Engineering Technology and Technopreneurship*, 2019.

[17] 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 International Conference on Informatics and Computing*, 2020.
[18] G. Indrawan, I. G. A. Gunadi, M. S. Gitakarma and I. K. Paramarta, Latin to Balinese Script transliteration: Lessons learned from the computer-based implementation, *Proc. of the International Conference on Software Engineering and Information Management*, 2021.
[19] 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 International Conference on Software Engineering and Information Management*, 2019.
[20] G. Indrawan, N. N. H. Puspita, I. K. Paramarta and S. Sariyasa, LBtrans-Bot: A Latin-to-Balinese Script transliteration robotic system based on Noto Sans Balinese font, *Indones. J. Electr. Eng. Comput. Sci.*, vol.12, no.3, pp.1247-1256, 2018.
[21] 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 International Conference on Innovative Research across Disciplines*, 2020.
[22] G. Indrawan, I W. Sutaya, K. U. Ariawan, M. S. Gitakarma, I G. Nurhayata and I K. Paramarta, A method for non-alphanumeric text processing on transliteration to the Balinese Script, *ICIC Express Letters*, vol.16, no.7, pp.687-694, 2022.
[23] I. M. Suatjana, *Bali Simbar*, 2022, http://www.babadbali.com/aksarabali/balisimbar.htm, Accessed on Aug. 01, 2022.
[24] I. M. Suatjana, *Bali Simbar Dwijendra*, Yayasan Dwijendra, Denpasar, 2009.
[25] I. B. A. Sudewa, *The Balinese Alphabet*, 2003, http://www.babadbali.com/aksarabali/alphabet.htm, Accessed on Aug. 01, 2022.
[26] I. G. K. Anom et al., *Balinese – Indonesian Dictionary with Its Latin and Balinese Script*, Balinese Language, Script, and Literature Development Agency, Bali Province, Denpasar, 2009.
[27] Y. Lee, Systematic homonym detection and replacement based on contextual word embedding, *Neural Process. Lett.*, vol.53, no.1, pp.17-36, 2021.
[28] H. Liu, M. Ma, L. Huang, H. Xiong and Z. He, Robust neural machine translation with joint textual and phonetic embedding, *Proc. of the Annual Meeting of the Association for Computational Linguistics*, 2020.
[29] K. Gorman, G. Mazovetskiy and V. Nikolaev, Improving homograph disambiguation with supervised machine learning, *Proc. of the International Conference on Language Resources and Evaluation*, 2019.
[30] J. Esling, *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*, Cambridge University Press, 1999.
[31] M. Smit and A. N. Al-Assimi, Cascade deep neural networks classifiers for phonemes recognition, *ARPN J. Eng. Appl. Sci.*, vol.15, no.7, pp.1664-1670, 2020.