

PENERAPAN ALGORITMA RULE BASE DENGAN PENDEKATAN HEXADESIMAL PADA TRANSLITERASI AKSARA BIMA MENJADI HURUF LATIN

(Implementation of Rule Base Algorithm with Hexadecimal Approach to Bima Aksara Transliteration to Latin Letter)

Arik Aranta*, Fitri Bimantoro, I Putu Teguh Putrawan
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: arikaranta@unram.ac.id, bimo@unram.ac.id, ipututeguhputrawan@gmail.com

Abstract

Bima Script is a text-based media exchange of information, a problem faced when wanting to learn the characters felt by the community, different forms of characters and contains basic rules that bind to the process of making words that are different from Latin letters so that when writing Aksara requires special expertise to remember the letters and the rules in punctuation. In this research, the author will develop an algorithm that can help in the activity of reading characters through transliteration media, which is devoted to the process of reading the Bima Script. The method used applies the machine learning concept that is applied to the literacy transliteration engine. With a string replacement algorithm that is optimized with hexadecimal number management derived from Unicode UTF-16 to support the level of complexity of the rules generated. After testing the algorithm the results obtained from the transliteration process by applying the algorithm got a system test that reached 90.64% of the 171 basis rules applied.

Keywords: Bima Aksara, String Replacement, Machine Learning, Hexadecimal Labeling, Transliteration.

*Penulis Korespondensi

1. PENDAHULUAN

Sejarah Indonesia tidak lepas dari proses pertukaran informasi dengan menggunakan media teks, media teks yang kita kenal saat ini adalah huruf latin atau Aksara Romawi dengan karakter A-Z. Karakter Aksara Latin yang dikenal masyarakat Indonesia oleh beberapa pendapat berasal dari masuknya kebudayaan Eropa ke Indonesia [1]. Sebelum masuknya bangsa Eropa ke Indonesia, masyarakat menggunakan berbagai macam media pertukaran informasi berbasis teks dengan nama Aksara. Kumpulan Aksara yang digunakan di Indonesia dikenal sebagai Aksara Nusantara. Aksara Nusantara yang digunakan di Indonesia berasal dari Aksara Plawa menurut Naomi, H dalam Erwin, A [2]. Saat ini Jumlah naskah Aksara Nusantara yang terdata di Indonesia sekitar +1000 naskah berada di Perpustakaan Nasional Republik Indonesia (PNRI), 5000 naskah berada +3500 Naskah berada Dayah Tanoh Abee, Seulimeum Aceh. +5000 berada di *Universiteits Bliibliothek*, Leinden, Belanda, +700 berada di *Muzium Islam Kuala Lumpur*, 6 berada di *South Africa Cultural Museum*, [1]. Jumlah tersebut merupakan sebagian kecil naskah yang berada di Indonesia. Penelitian dalam proses mempelajari

naskah Aksara tersebut, menjadi sebuah kajian atau topik yang menarik untuk diteliti, dikarenakan pada naskah tersebut tersimpan berbagai Informasi terkait sejarah perjuangan, kebudayaan, sejarah terkait persebaran Agama Bangsa Indonesia.

Permasalahan yang dihadapi oleh Bangsa Indonesia saat ini adalah minimnya penggunaan Aksara Daerah pada proses pertukaran informasi oleh masyarakat, dalam hal ini peran Pemerintahan Indonesia diperlukan untuk membangkitkan penggunaan Aksara untuk kehidupan sehari-hari. Salah satu Daerah yang telah mengeluarkan kebijakan untuk membangkitkan penggunaan Aksara Daerah adalah Pemerintahan Provinsi Bali dengan mengeluarkan peraturan dalam penggunaan Aksara pada fasilitas, seperti penulisan nama jalan, prasasti peresmian gedung, sarana pariwisata, dan beberapa fasilitas umum lainnya seperti yang tersebut pada Peraturan Gubernur Bali [3].

Untuk mendukung kegiatan Pemerintah terkait meningkatkan penggunaan Aksara pada kehidupan sehari-hari, salah satu upaya yang dapat dilakukan adalah mendigitalisasi Aksara, agar dapat masuk pada perkembangan dunia teknologi informasi. Penggunaan Teknologi informasi saat ini berdasarkan survey yang

dilakukan 65,2% responden di wilayah urban dan 26,11% di wilayah rural telah menggunakan internet untuk berbagai kepentingan di Indonesia [1].

Permasalahan yang dihadapi dalam proses penggunaan Aksara saat ini adalah terkait pada proses transliterasi. Transliterasi adalah kegiatan merubah suatu huruf menjadi huruf lainnya, beberapa permasalahan terkait transliterasi pada penulisan Aksara Nusantara adalah berbedanya susunan suatu huruf ketika diterapkan pada huruf lainnya, sehingga memerlukan keahlian khusus dalam menulis Aksara [5][6].

Berdasarkan data pengguna internet tersebut, maka diperlukannya sentuhan teknologi informasi pada Aksara Nusantara agar dapat kembali digunakan menjadi media komunikasi dimasyarakat khususnya masyarakat Indonesia. Beberapa penelitian terkait digitalisasi Aksara Nusantara termasuk proses transliterasi huruf latin Indonesia menjadi Aksara, telah dilakukan diantaranya penemuan metode hingga pengembangan beberapa aplikasi dan penelitian terkait Aksara Nusantara [5][6][7][8][9][10] [11].

Pengembangan media pembelajaran terkait Aksara juga telah dikembangkan dengan melibatkan penggunaan *smart phone* sebagai media pembelajaran [12]. Dengan menggunakan pendekatan media teknologi informasi diharapkan dapat membangkitkan gairah penggunaan Aksara di kalangan masyarakat untuk kembali mempelajari Aksara Nusantara diataranya adalah Aksara Bima.

Berangkat dari permasalahan tersebut dan belum ditemukannya proses transliterasi digital perubahan Aksara Bima menjadi huruf latin Indonesia, sehingga membuat diperlukannya penelitian terkait proses transliterasi tersebut. Penelitian ini membahas sebuah metode *string replacement* dengan optimasi heksadesimal untuk proses transliterasi dari huruf Aksara Bima menjadi huruf latin sehingga dapat memudahkan proses membaca huruf Aksara Bima yang terdapat pada sebuah text menjadi huruf latin.

2. TINJAUAN PUSTAKA

Beberapa penelitian terkait proses transliterasi telah dilakukan dimana terdapat beberapa metode dalam proses perubahan dari huruf latin menjadi Aksara, berikut akan diuraikan penelitian terkait proses transliterasi pada Aksara Nusantara.

Penelitian dengan judul *Utilization of Hexadecimal Numbers In Optimization of Balinese Transliteration String Replacement Method* [5]. Pada jurnal ini dibuat sebuah aplikasi transliterasi dengan menerapkan metode penggantian *string*. Dalam prosesnya, tulisan

yang bertipe data *string* akan dirubah menjadi angka heksadesimal sebagai kode unik dari masing-masing tiap karakter agar terhindar dari kesamaan antara aturan, masing-masing karakter Bali yang terdiri dari 217 karakter. Akurasi aplikasi berbasis android yang dihasilkan ini yaitu 92,72% dari 3915 data tes yang digunakan sebagai sampel penelitian, Penelitian tersebut yang dijadikan acuan dalam pengembangan metode *string replacement* yang diterapkan dikarenakan memiliki kesamaan dalam pemanfaatan bilangan heksadesimal, namun pada penelitian ini karakter heksadesimal yang digunakan belum melalui proses penjumlahan antar bilangan sehingga apabila diperlukan sebuah *update* aturan penulisan, pengembang diharuskan mencari satu-persatu karakter pada tabel heksadesimal, dan pada penelitian ini proses penjumlahan bilangan heksadesimal diterapkan untuk proses pencarian *rule base* yang diterapkan pada proses pembuatan alamat bilangan heksadesimal pada karakter Aksara.

Penelitian kedua berjudul Pembuatan perangkat lunak transliterator Aksara latin ke Aksara Bali dan Aksara Bali ke Aksara latin [11], pada penelitian tersebut menjelaskan sebuah proses mengalih tuliskan Aksara Bali yang terdapat pada lembaran lontar, pada sebuah media digital. Pada penelitian tersebut menerapkan proses transliterasi Aksara Bali menjadi huruf latin dan sebaliknya, proses transliterasi yang berjalan pada penelitian ini adalah dengan menerapkan aturan yang dibangun berdasarkan sifat dari karakter yang di transliterasikan, mengingat proses tersebut diperlukan karena Aksara Bali memiliki beberapa karakter khusus yang disebut *pasang pageh*, dimana karakter tersebut memiliki taturan penulisan tersendiri atau mutlak yang tidak dapat dieksekusi dengan aturan penulisan Aksara Bali pada umumnya.

Penelitian ketiga adalah Analisis *Crowd Feedback* Aplikasi *Mobile* Pembelajaran Transliterasi Tulisan Latin Ke Aksara Bali. Pada jurnal ini dibuat sebuah aplikasi transliterasi tulisan latin ke Aksara Bali dengan acuan dokumen "*The Balinese Alphabet*" sebagai aturan transliterasi. Terdapat jumlah instalasi aplikasi sebanyak 32 ribu instalasi dan (152 *rating*) dan 57% dari total review mendapatkan penerimaan di masyarakat relatif baik [13].

Penelitian keempat dengan judul Transliterasi Huruf Latin Bahasa Indonesia ke Aksara Jawa Menggunakan Metode *Snakecut* [14]. Pada jurnal ini membahas tentang transliterasi Aksara Jawa berbasis *image processing*. Gambar yang diambil akan dikonversi melalui aplikasi Microsoft Visual Studio 2010 dengan *library* Emgu CV sebagai pengolahan citra dan

dalam penelitian ini mendapat tingkat akurasi 85% dari hasil pengujian percobaan yang berhasil 17 dari 20 sampel data uji.

Penelitian kelima dengan judul *Balinese Latin Text Becomes Aksara Bali Using Rule Base Method* [8]. Pada jurnal ini, dilakukan pembuatan sistem yang mampu mengubah teks latin menjadi Aksara Bali. Sistem ini melakukan transliterasi dengan metode *rule base*. Dalam proses transliterasi yang dilakukan mendapatkan hasil transliterasi 90,67% dengan proses pengujian menulis huruf latin dan dilakukan proses pencocokan dengan citra lontar.

Penelitian keenam dengan judul implementasi Komputasi Paralel untuk Optimalisasi Komputasi pada Aplikasi Transliterasi Huruf Latin ke Aksara Jawa, penelitian tersebut membahas sebuah metode untuk proses komputasi pada aplikasi transliterasi Aksara Jawa. Sehingga dengan penerapan algoritma tersebut dapat menghemat proses komputasi pada proses transliterasi menggunakan metode *decision tree* dan mendapatkan hasil dalam proses pengurangan lama proses komputasi transliterasi Aksara rata-rata 90.75% [14].

Penelitian ketujuh dengan judul Program Transliterasi Antara Aksara Latin dan Aksara Jawa dengan Metode FSA, dalam penelitian ini mengembangkan sebuah proses transliterasi Aksara Jawa dengan metode FSA (*Finistate Automata*), dengan dua arah transliterasi dengan input latin dan menghasilkan output Aksara Jawa, dan proses input Aksara menjadi huruf latin. Dimana dalam proses pengujian didapat sebuah kesimpulan, yang mentakan dalam sebuah proses transliterasi terdapat sebuah model linier yang menunjukkan jumlah karakter dengan tingkat kerumitan kata, berbanding *horizontal* dengan waktu eksekusi program [7].

Dalam penelitian kedelapan dengan judul *Algorithm to Avoid Overlapping Vowel Signs in Latin to Balinese Script Transliteration Method*[6], melakukan pembahasan yang lebih mendalam terkait transliterasi Aksara Bali, dimana dalam sebuah proses transliterasi Aksara Bali terdapat suku kata yang terletak diatas karakter utama, hal ini tentunya akan menjadi permasalahan apabila terdapat 2 karakter yang berada daam sebuah karakter utama, menggunakan algoritma tersebut permasalahan *overlapping vowel* dapat diatasi.

Berangkat dari beberapa referensi tersebut belum ditemukanya proses transliterasi Aksara Bima menjadi huruf latin, maka dalam hal ini penulis melakukan proses penelitian menerapkan metode *rule base* yang dioptimalkan dengan penggunaan nilai heksadesimal

yang berangkat dari penelitian sebelumnya [5], dengan mengembangkan proses pengalamatan penjumlahan karakter heksadesimal, bertujuan untuk membuat sebuah hubungan korelasi antara karakter utama dengan karakter perubahanya, yang sesuai dengan aturan penulisan. Hal tersebut dapat menyelesaikan permasalahan yang dihadapi dalam transliterasi Aksara Bima menjadi huruf latin kembali yang sesuai dengan aturan penulisan, dan dapat membuat sebuah proses pencarian *rule base* pada setiap karakter dari hasil penjumlahan bilangan heksadesimal pada karakter.

2.1. Aksara Bima

Aksara Bima merupakan Aksara yang telah digunakan masarakat Bima sejak abad XIV Masehi, terdapat kontroversi terkait keberadaan Aksara Bima sebelum dilakukanya kajian dan penelitian yang dilakukan oleh (Hj. Siti Maryam R. Salahuddin) , terhadap naskah yang ditemukanya pada salah satu naskah perjalanan seorang peneliti Balanda, H. Zollinger yang pernah melakukan perjalanan ke Bima dan Sumbawa Indonesia, pada tahun 1847 dengan judul dokumen Alfabet Bima yang telah hilang [1]. Pada tahun 1990-1991 seorang Guru Besar dari Universitas Leinden, Ahli bahasa dan Aksara Bugis, Belanda, bernama J.Noorduyn melakukan klarifikasi terhadap naskah lontar yang diasumsikan olehnya sebagai Aksara Bugis, namun pernyataan tersebut terbantahkan sendiri olehnya, dikarenakan peneliti tersebut yang merupakan ahli Aksara Bugis yang tidak dapat membaca naskah tersebut, dan setelah melewati roses penelitian dan disesuaikan dengan dasar abjad yang di temukan oleh (Hj. Siti Maryam) pada perpustakaan RI tahun 1987, dan dikaji satu persatu oleh J.Noorduyn di Leinden, dan dibawa kembali lagi ke Indonesia untuk dikaji oleh (Hj. Siti Maryam), maka terangkailah sebuah kalimat berbahasa Bima. Aksara Bima dideklarasikan pada tanggal 28 Juli 2007, pada acara penutupan Simposium International Penaskahan XI, yang dilaksanakan di Bima. Namun pada Simposium tersebut belum memberikan informasi secara utuh terkait Aksara Bima sehingga masih diperlukanya kajian serius terhadap kandungan Aksara Bima[1].

Aksara Bima yang terdata pada penelitian ini berjumlah 26 karakter, pada acuan pustaka Aksara Bima terdiri dari dua huruf. yang pertama adalah karakter A-Y dimana terdapat pengecualian terhadap karakter Q, V, X dan Z. Karakter tersebut tidak ditemukan bentuk hurufnya, namun terdapat perubahan fungsi terhadap karakter Q yang dinilai menjadi karakter K, seperti penulisan kata "*mutlaq*" menjadi "*mutlak*" berikut merupakan karakter Aksara

Bima yang terdata pada buku Aksara Bima Peradaban Lokal yang Sempat Hilang [1].

TABEL I. BENTUK AKSARA BIMA PERHURUF

Cara Baca	Aksara	Cara baca	Aksara
A		LA	
BA		MA	
CA		NA	
DA		PA	
FA		RA	
GA		SA	
HA		TA	
JA		WA	
KA		YA	

Tabel I. melampirkan bentuk *font* Aksara Bima, Aksara Bima dalam bentuk digital tersebut berasal dari komunitas pengembang Aksara Nusantara [14]. Aksara yang dimunculkan bersal dari karakter A – YA karakter utama ini yang membentuk kalimat Aksara Bima, kalimat tersebut merupakan kalimat yang bersal dari suara konsonan. Sehingga apabila akan diucapkan dengan lisan maka perlu ditambaknya karakter vokal seperti a, i, u, e, o. Pada penulisan Aksara Bima huruf vokal a adalah karakter yang berdiri sendiri tanpa harus memerlukan bantuan karakter konsonan seperti karakter pada tabel I.

Pada Tabel II akan dimunculkan karakter vokal i, u, e, o, yang dipasangkan dengan karakter “a”. Proses pemunculan karakter vokal dapat dilihat pada tabel 2 sebagaimana berikut.

TABEL II. BENTUK VOKAL AKSARA BIMA

Cara Baca	Aksara	Cara baca	Aksara
a		i	
u		e	
o			

Pada tabel II ditampilkan karakter vokal yang melekat pada karakter utama, seperti yang terdapat pada tabel I. Seperti pada tabel II dalam penulisan karakter dengan penambahan vokal askara Bima pada karakter percontohan A, ter susun penulisan A, Ai, Au,

eA, Ao, karakter A tersebut dapat digantikan dengan karakter konsonan lainnya dengan aturan yang sama. Dalam penulisan karakter dengan penambahan suara vokal terdapat permasalahan yaitu karakter e ditulis dengan penulisan eA dan bukan Ae seperti karakter lain dalam Tabel II hal ini berlaku pada seluruh karakter konsonan.

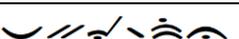
Pada Tabel III ditampilkan Aksara Bima dalam bentuk sengau, karakter sengau adalah karakter yang pada proses pelafalaya memerlukan pemampatan nafas pelafalnya, karakter sengau dimunculkan dalam Aksara Bima terdiri dari 3 karakter dalam sebuah simbol. Terdapat beberapa karakter sengau yang apabila ditemukan mengembalikan pada karakter utama, seperti pada Tabel I diantaranya karakter MBA yang akan berubah menjadi BA, Karakter NDA yang akan berubah menjadi DA, karakter NGGA yang akan berubah menjadi GA.

TABEL III. SAMPLE BENTUK HURUF SENGAU AKSARA BIMA

Cara Baca	Aksara	Cara baca	Aksara
MBA		NGGA	
NCA		NTA	
NDA		MPA	
NGA			

Berikutnya Pada Tabel IV ditampilkan Aksara Bima dalam bentuk sebuah kalimat. terbentuknya suatu kalimat terdiri dari huruf Aksara Bima yang disusun dengan aturan tertentu.

TABEL IV. PENULISAN AKSARA BIMA DALAM BENTUK KATA

No	Latin	Symbol
1	Niat	
2	Ndai Ma Wa'ana	
3	Makanggorina	

Pada Tabel IV merupakan perubahan kata latin menjadi askara Bima, dimana karakter spasi dan tanda baca tidak dikenali pada Aksara yang ditampilkan pada Tabel [1]. Proses tersebut akan memunculkan sebuah permasalahan khusus apabila akan

menerapkan sebuah proses transliterasi perubahan dari huruf Aksara menjadi huruf latin.

2.2. Angka Heksadesimal

Angka heksadesimal Merupakan bilangan yang terdiri dari karakter A-F dimana karakter tersebut adalah kombinasi bilangan angka 0-9, dan huruf A-F, sehingga jumlah dari bilangan tersebut adalah 16 karakter. Bilangan hexadesimal pada proses ini mengadopsi format bilangan unicode UTF-16 dengan menghapus \u, dikarenakan proses penjumlahan heksadesimal hanya mengenali karakter 0-9 dan A-F. Belum terdatanya karakter Aksara Bima pada unicode membuat proses penelitian ini melakukan proses adopsi karakter unicode yang berasal dari bilangan unicode dari karakter latin yang dimunculkan setelah konversi menjadi huruf latin. Heksadesimal memiliki jumlah karakter lebih banyak 6 angka dari bilangan desimal membuat karakter heksadesimal dapat menyingkat penulisan bilangan angka dalam jumlah bilangan ribuan pada bilangan desimal. Bilangan desimal dengan angka 6015 dalam heksadesimal akan menjadi 177F. Beberapa Penelitian yang memanfaatkan bilangan heksadesimal sebagai pilihan untuk kolaborasi antar data yang lebih kompleks, diantaranya pada macam permasalahan seperti proses pengenalan karakter huruf dan simbol [16]. Penerapan bilangan heksadesimal telah umum digunakan untuk proses pengalamatan karakter dimana beberapa karakter Aksara telah mendapat code unicode dengan bilangan heksa desimal [16]. Dalam penelitian ini deretan angka yang dihasilkan akan memanggil karakter yang di alamatkan seperti pada karakter yang terdata pada unicode [16]. Dimana bilangan heksadesimal berperan pada proses pengalamatan karakter Aksara Bima dan ditambah dengan proses penjumlahan bilangan heks desimal untuk proses pengalamatan karakter baru sehingga proses penulisan *rule base* dapat menelusuri korelasi antara satu karakter dengan karakter lainnya pada Aksara Bima sehingga penyusunan aturan dapat lebih variatif dan lebih kompleks.

2.3. Metode Pergantian String

Metode pergantian *string* atau *string replacement* adalah metode yang digunakan dalam proses pemrograman untuk merubah suatu string menjadi string yang lain.

```
input = "Belajar Aksara Bima itu sulit"
a=(input
#angka
.replace("sulit", "sangat mudah"))
print(a)
```

Belajar Aksara Bima itu sangat mudah

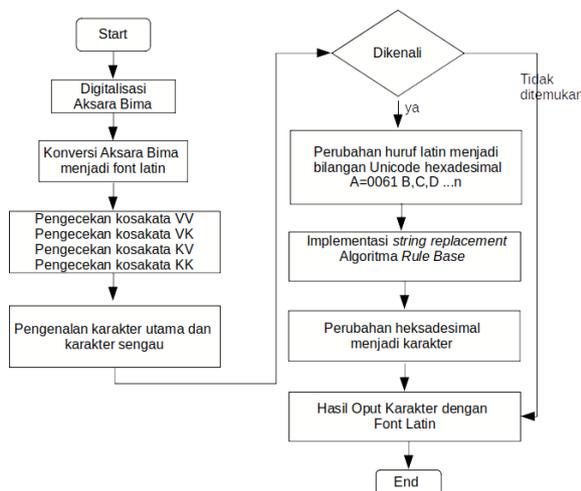
Gambar 1. Sintaks *String Replacement*

Pada Gambar 1 merupakan sebuah proses *string replacement* pada bahasa pemrograman Python 3, pada proses tersebut terjadi perubahan pada karakter yang di inputkan dengan karakter yang dikalikan, dimana karakter dengan masukan "Belajar Aksara bima itu sulit" dan terjadi *replacement* kata "sulit", menjadi kata "sangat mudah" maka *output* yang dihasilkan adalah "Belajar Aksara Bima itu sangat mudah".

3. METODE PENELITIAN

Proses implementasi metode pada penelitian ini dipengaruhi oleh beberapa proses, diantaranya adalah proses pengalamatan karakter dengan bilangan heksadesimal, dan proses penerapan akan membuat sebuah alur program sebagaimana *flow chart* Pada Gambar 2. Dalam *flow chart* aplikasi terdiri dari beberapa tahapan. Proses awal adalah menyiapkan *file* Aksara Bima yang akan ditransliterasikan, pada case ini penulis menggunakan buku acuan sebagai validator [1], sebagai *file* uji aplikasi langkah berikutnya adalah menyalin tulisan menggunakan karakter dengan *font* Aksara Bima, *font* Aksara Bima yang digunakan adalah *font* dari Aksara Bima Mbojo yang berasal dari komunitas pengembang Aksara Nusantara [14], langkah ini sekaligus memudahkan proses pengenalan karakter dan menampilkan unicode dari karakter Aksara Bima yang dirubah menjadi karakter latin. Langkah kedua karakter dengan *font* Bima Mbojo dirubah menjadi *font* latin (*Times New Roman*) penggunaan *font* tersebut dapat digantikan oleh font lain seperti *calibri*, *arial* atau font latin lainnya, yang telah umum digunakan. Langkah Ketiga adalah tahap perubahan huruf latin menjadi bilangan heksadesimal pada tahap ini penulis menggunakan unicode *converter online* untuk mendapatkan nilai UTF-16 setiap karakter latin, pada proses ini perubahan karakter latin hasil konversi tahap dua diberikan alamat heksadesimal guna proses implementasi metode *rulebase*. Langkah keempat adalah proses pengecekan apakah karakter mendapatkan alamat bilangan heksadesimal atau tidak, karakter yang tidak mendapatkan alamat bilangan heksadesimal maka akan langsung dimunculkan kembali dengan *font* latin pada proses *output* sistem.

Langkah kelima adalah proses pengecekan karakter, K (Konsonan) dan V (Vokal) pada proses pengenalan karakter, terjadi sebuah seleksi apakah karakter yang akan dieksekusi adalah karakter vokal atau konsonan. Proses ini diperlukan untuk proses persiapan penerapan *rulebase*, apabila terapat sebuah karakter, dengan suatu contoh karakter N, maka sistem akan mencocokkan dengan karakter didepan atau dibelakang, karakter N tersebut. Untuk mengetahui karakter N tersebut termasuk dalam karakter N dengan vokal seperti NA atau karakter N dalam huruf mati. Berikut seterusnya hingga karakter yang memasuki proses *rule base* dikenali apakah KK, KV, VK, VV. Tahap ini diperlukan guna mempelajari karakter yang dimaksud termasuk dalam jenis karakter Bima yang telah didata pada tabel *rule base* dengan nomor heksadesimal tertentu .



Gambar 2. Flow Chart Sistem

Langkah keenam Penerapan algoritma *rule base*, detail algoritma *rule base* dapat dilihat pada sub-point 3.2. dimana pada proses ini karakter akan dikembalikan sesuai dengan urutannya. Langkah ketujuh adalah proses untuk merubah heksadesimal yang telah membentuk suatu deret pola angka heksadesimal, dimana pola tersebut berisikan susunan karakter yang digunakan untuk memunculkan sebuah kalimat yang akan dirubah menjadi karakter latin yang dapat terbaca.

Langkah kedelapan adalah proses terakhir dalam sebuah proses transliterasi Aksara menjadi huruf latin, dimana pada proses ini karakter yang telah berubah menjadi *symbol* ditampilkan pada sistem dalam bentuk sebuah kata utuh dengan *font* latin seperti yang telah disebutkan.

3.1. Pengalamatan Dengan Heksadesimal

Proses pengalamatan dengan bilangan heksadesimal, adalah proses dimana setiap karakter memiliki kode unik yang di dapat dari karakter yang diinisialkan sebagai mana contoh karakter “a” memiliki *code* unicode UTF-16 adalah \u0061, karakter unicode tersebut akan dijadikan identitas bilangan heksadesimal pada proses transliterasi, ada tahapan yang harus dilakukan agar karakter dapat dikelola pada proses pengimplementasian *rulebase*. Bilangan heksadesimal yang digunakan, adalah Bilangan heksadesimal yang terdiri dari 4 digit bilangan, dalam proses ini karakter heksadesimal yang digunakan harus lah terinput dengan jumlah karakter yang sama, perbedaan jumlah krakter bilangan yang berada dalam 1 *symbol* akan dapat merubah urutan sebuah proses transliterasi, apabila terdapat karakter yang muncul dengan kode \u0061 maka tidak dapat dituliskan dengan karakter 61 namun harus memunculkan dua digit didepanya dan menghapus \u sehingga menjadi 0061. Adapun hasil pemetaan yang dilakukan terhadap karakter Aksara Bima seperti pada Tabel V.

TABEL V. PENGALAMATAN KARAKTER PERHURUF

Aksara	K.Sym	Hex	Aksara	K.Sym	Hex
a	a	0061	la	l	006C
ba	b	0062	ma	m	006D
ca	c	0063	na	n	006E
da	d	0064	pa	p	0070
fa	f	0066	ra	r	0072
ga	g	0067	sa	s	0073
ha	h	0068	ta	t	0074
ja	j	006A	wa	w	0077
ka	k	006B	ya	y	0079

Pada tabel V merupakan proses deklarasi dari Tabel I dimana karakter Aksara Bima pada Tabel I tersebut diberikan pengalamatan bilangan heksadesimal, pada sistem setiap huruf konsonan yang ditulis maka akan secara otomatis karakter Aksara

Bima mendapat vokal karakter A, sebagai proses yang memudahkan pengucapan.

TABEL VI. PENGALAMATAN KARAKTER VOKAL

Aksara	K.Sym	Hex	Aksara	K.Sym	Hex
a	a	0061	e	e	0065
i	i	0069	o	o	006f
u	u	0075			

Tabel VI merupakan hasil pengalaman karakter vocal dengan bilangan heksadesimal. Tabel VII merupakan proses pengalaman karakter dengan huruf vokal setelah huruf konsonan. pada tahap ini seluruh huruf konsonan dipasangkan dengan huruf vokal diataranya, a, i, u, e, o dan diujicobakan pada karakter B sampai karakter Y seperti pada Tabel I. dimana pada Tabel VII merupakan sebuah *sample* yang mewakili proses pembentukan sebuah alamat pada karakter konsoan yang bertemu huruf vokal.

TABEL VII. SAMPLE KARAKTER KONSONAN BERTEMU VOKAL

Aksara	Rule	Hex	Aksara	Rule	Hex
ba	0062+0061	00C3	be	00C3+0065	0128
bi	00C3+0069	012C	bo	00C3+006F	0132
bu	00C3+0075	0138			

Pada Tabel VII adalah proses menghasilkan bilangan heksadesimal apabila bertemu vokal didepanya, seluruh karakter konsonan bertemu vokal i, u, e, o menggunakan penjumlahan dari karakter B bertemu A. sehingga ketika akan memunculkan karakter "Bi" hasil perhitungannya akan seperti pada Persamaan (1). hal tersebut berlaku sama pada konsep perhitungan karakter konsonan seterusnya.

$$bi = 0062 + 0061 + 0069 = 00C3 \quad (1)$$

Pada Tabel VIII merupakan proses pembuatan identitas pada karkater sengau, proses pembuatan identitas pada karakter sengau merupakan hasil dari penjumlahan bilangan hexadesimal pada karakter sebelumnya.

TABEL VIII. KARAKTER SENGAU AKSARA BIMA

No	Aksara	Rule	Hex
1	mba	00CE+00C3	019C
2	mbi	00CE+012C	01FA
3	mbu	00CE+0138	0206
4	mbe	00CE+0128	01F6
5	mbo	00CE+0132	0200
6	nca	00CF+00C4	0193
7	nci	00CF+00CC	0191
8	ncu	00CF+00D8	01A7
9	nce	00CF+00C8	0197
10	nco	00CF+00D2	0191
11	nda	00CF+00C5	0194
12	ndi	00CF+00CD	019C
13	ndu	00CF+00D9	01A8
14	nde	00CF+00C9	0198
15	ndo	00CF+00D3	01A2
16	nga	00CF+00C8	0197
17	ngi	00CF+0131	01A0
18	ngu	00CF+00DC	01AB
19	nge	00CF+00CC	019B
20	ngo	00CF+00D6	01A5
21	ngga	00CF+0190	025F
22	nggi	00CF+01F9	02C8
23	nggu	00CF+0205	02D4
24	ngge	00CF+01F5	02C4
25	nggo	00CF+01FF	02CE
26	nta	00CF+00D5	01A4
27	nti	00CF+00DD	01AC
28	ntu	00CF+00E9	01B8
29	nte	00CF+00D9	01A8
30	nto	00CF+00E3	01A2
31	mpa	00CE+00D1	019F
32	mpi	00CE+00D9	01A7
33	mpu	00CE+00E5	01B3
34	mpe	00CE+00D5	01A3
35	mpo	00CE+00DF	01AD

Proses penjumlahan yang terjadi mengikuti jumlah karakter dimana terdapat karakter dengan jumlah 3 karakter hingga karakter dengan jumlah 4 karakter.

Persamaan (2- 4) yang merupakan proses untuk menghasilkan sebuah identitas bilangan pada karakter sengau, yang berasal dari hasil penjumlahan bilangan heksadesimal.

$$mba = maba = (006D + 0061) + (0062 + 0061) = 019C \quad (2)$$

$$mbi = mabai = (006D + 0061) + ((0062 + 0061) + 0069) = 01FA \quad (3)$$

$$nggi = nagagai = (006A + 0061) + ((0067 + 0061) + (0067 + 0061) + 0069) \\ nggi = 02C4 \quad (4)$$

Pada Persamaan (2) menjelaskan proses untuk mendapatkan identitas heksadesimal karakter “mba”, karakter tersebut dijabarkan dengan aturan semua karakter vokal yang muncul akan ditambahkan karakter “a”, dan setelah mendapatkan karakter vokal “a” maka semua karakter dirubah menjadi bilangan heksadesimal. setelah menjadi bilangan heksadesimal maka setiap karakter akan dijumlahkan dan mendapatkan hasil yang dijadikan identitas oleh karakter yang dimaksud, sehingga karakter “mba” memiliki identitas “019C”, dan hal ini berlaku untuk rumus Persamaan (3 dan 4).

3.2. Penerapan Algoritma *rulebase* dengan *String Replacement*

Algoritma *rulebase* adalah algoritma yang digunakan untuk proses pemberian hukum bacaan dari Aksara Bima. apabila sebuah karakter untuk mendapatkan sebuah identitas, dilakukan sebuah proses perhitungan dari karakter yang terkandung. Proses pembuatan *rulebase* merupakan proses dimana bilangan heksadesimal yang telah dimiliki oleh karakter tersebut akan dikelola untuk mendapatkan aturan yang sesuai dengan aturan yang ditetapkan.

Sebuah permasalahan pada Tabel IX akan menjelaskan bagaimana algoritma *rulebase* tersebut dioperasikan.

TABEL IX. IMPLEMENTASI RULE BASE DUA KATA

Aksara	Key.Sym	Latin
	bim	bima

Seperti pada skema Tabel IX terdapat sebuah Aksara Bima yang didapat dari hasil menulis pada *text editor*, pada tahap ini peneliti memunculkan *font* Aksara bima pada *libre office*, pada proses

transliterasi dari huruf Aksara Bima, yang akan dirubah menjadi huruf latin. Kata dalam *font* Aksara Bima yang telah ditulis pada *text editor* dan di rubah menjadi font *calibri*, hanya menampilkan karakter *b, i, m*. sedangkan huruf latin yang seharusnya dimunculkan adalah kata “bima”, sehingga pada proses ini memerlukan sebuah penerapan algoritma *rule base*, adapun proses penerapan *rule base* memanfaatkan pengelolaan dengan teknik *string replacement* pada bilangan hexadesimal, dapat dilihat pada formula yang dioperasikan didalam sistem adalah sebagaimana berikut.

$$bim = ((0062 + 0061) + 0069)(006D + 0061) = 012C00CE \quad (5)$$

$$bim = 012C00CE = bima \quad (6)$$

Pada Persamaan (5) merupakan sebuah tahap pertama dalam penginputan karakter pada sistem, dimana karakter “bim” dirubah menjadi sebuah bilangan hexadesimal selanjutnya bilangan heksadesimal melalui proses filterasi untuk membuat sebuah *rule base*, dimana setiap karakter konsonan akan ditambah karakter A (0061), proses ini diperlukan untuk membuat karakter konsonan dapat diucapkan secara normal dan tidak diucapkan dalam huruf mati. Langkah berikutnya, terdapat penggabungan 0062 dengan 0061 dimana pada tahap ini 2 bilangan tersebut di lakukan proses penjumlahan, proses penjumlahan dilakukan manual diluar sistem oleh penulis, dan hasil penjumlahan tersebut digunakan untuk memberikan alamat pada Aksara Bima, dari hasil penjumlahan didapat sebuah alamat dengan angka hexadecimal 00C3, yang merupakan alamat dari simbol “ba” pada tahap ini karakter membentuk sebuah tulisan “baima”, sehingga pada tahap berikutnya terjadi proses perjumlahan 00C3 dengan 0069 dimana membentuk karakter 012C dengan output karakter “bi” dan hasil penjumlahan karakter 006D dan 0061 dengan karakter “m” memunculkan bilangan 00CE yang merupakan alamat dari karakter “ma” sehingga apabila dikembalikan pada karakter yang miliki alamat 012C00CE menjadi kata “bima” dengan hasil tersebut proses dihentikan dengan menampilkan kata sesuai dengan harapan. adapun hasil implemntasi pada sistem akan terlihat hasilnya sebagaimana Gambar 3.

Pada Gambar 3 merupakan sampel implementasi pada sistem, yang telah diuraikan pada tahapan tahapan sebelumnya.

```
In [2]: str = "bim"
a=(str
#huruf utama
.replace("b", "ba")
.replace("i", "i")
.replace("m", "ma")
#Karakter Sengau
#huruf ke heksa
b=(a
#Sengau ke heksa
#Utama ke heksa
.replace("a", "0061")
.replace("b", "0062")
.replace("i", "0069")
.replace("m", "006D")
)
#rule base
c=(b
#reposisi "e"
#rule karakter utama
.replace("006200610069", "012C")#bai
.replace("006D0061", "00CE")#ma
#heksa ke huruf)
d=(c
.replace("012C", "bi")
.replace("00CE", "ma"))
#hasil
print (d)

bima
```

Gambar 3. Implementasi pada Sistem

Pada proses tersebut terlihat, inputan kata “bim” yang bersal dari proses konversi *font* Aksara Bima Mbojo, yang berikutnya dirubah menjadi *font calibri*, dan hasil konversi tersebut di inputkan pada algoritma yang telah disusun sehingga inputan akan diproses dan menghasilkan kata “bima” sesuai dengan aturan yang dimiliki oleh Aksara Bima.

Sampel *Case 2*, pada sampel *case 2* permasalahan yang digunakan dan validasi data bersumberkan dari buku acuan [1].

BAHASA BIMA	AKSARA BIMA
Makanggorina	u // d i a
Ndai Ma Wa'ana	u m m m m
Ngarana	l a a
Sambeya	o i n x

Gambar 4. Data Validasi

Gambar 4 adalah model data validasi yang digunakan oleh sistem untuk menguji proses transliterasi data, validasi ini di pilih dari buku acuan dikarenakan terdapat penggunaan penulisan Aksara utama, Aksara sengau, penggunaan huruf vokal, dan penerapan tanda baca. Setelah data validasi disiapkan maka langkah berikutnya adalah melakukan proses digitalisasi Aksara Bima yang akan di proses, proses digitalisasi Aksara Bima menerapkan proses perubahan font Bima Mbojo pada penelitian ini

penulis menggunakan perubahan menjadi *font calibri* dan didapatkan hasil sebagaimana Tabel X berikut.

TABEL X. IMPLEMENTASI RULE BASE DUA KATA

No	Font Aksara Bima	Font latin	Bahasa
1		mkgorin	Makanggorina
2		daimwan	Ndai Ma Wa'ana
3		Nrn	Ngarana
4		seby	Sambeya

Pada Tabel X merupakan contoh Aksara yang telah melalui proses digitalisasi Aksara menggunakan akasara Bima Mbojo, dimana dapat dilihat pada kolom tabel font Aksara Bima, pada kolom font Aksara Bima penulisan telah ditulis sesuai dengan aturan pada sesuai dengan dokumen validasi seperti pada Gambar 4. dan apabila dirubah *font* latin seperti pada kolom *font* latin calibri, hasil perubahan *font* tersebut masih memiliki perbedaan dengan bahasa Bima sebenarnya, maka pada proses berikutnya adalah melakukan pemrosesan pada sistem transliterasi yang telah di rancang seperti pada Gambar 5.

```
str = "mkgorin, daimwan, Nrn, seby "
a=(str
#huruf utama
.replace("a", "a")
.replace("b", "ba")
.replace("c", "ca")
.replace("d", "da")
.replace("e", "e")
.replace("f", "fa")
.replace("g", "ga")
.replace("h", "ha")
.replace("i", "i")
.replace("j", "ja")
#dan seterusnya
#Karakter Sengau
.replace("N", "Na"))
#penyusunan sesuai abjad
```

Gambar 5. Pemrosesan Karakter Utama dan Sengau

Gambar 5 menjelaskan proses *string replacement* terhdap 4 buah inputan “mkgorin, Nrn, seby”. Dimana setiap karakter konsonan yang terinput akan ditambahkan karakter vokal “a”, setelah penjabaran karakter utama dilanjutkan dengan karakter sengau, yang pada proses implementasinya menambahkan vokal karakter “a”. Hal tersebut dilakukan secara

berulang dengan urutan sesuai abjad, data yang diinputkan berdasarkan urutan huruf dan karakter Aksara Bima, karakter sengau dalam proses ini menerapkan pengurutan sesuai abjad.

```
#huruf ke heksa
b=(a
#Karakter Utama
.replace("a", "0061")
.replace("b", "0062")
.replace("c", "0063")
.replace("d", "0064")
.replace("e", "0065")
.replace("f", "0066")
.replace("g", "0067")
.replace("h", "0068")
.replace("i", "0069")
.replace("j", "006A")
#dan seterusnya
#Karakter Sengau
.replace("N", "0132"))
#penyusunan sesuai abjad
```

Gambar 6. Perubahan Huruf mejadi Heksadesimal

Gambar 6 menjelaskan proses perubahan dari karakter latin menjadi bilangan heksadesimal, bilangan tersebut berasal dari hasil penjabaran pada tabel karakter utama, dan karakter sengau.

```
#rule base
c=(b
#1. karakter vokal i, u, e, o
.replace("006200610069", "012C")#"bai" menjadi "bi"
#berlaku pada "bai, bau, bae, bao"
.replace("00670061006F", "0137")#"gao" menjadi "go"
.replace("007200610069", "013C")#"rai" menjadi "ri"
#dilakukan pada karakter b-y
.replace("006500620061", "0128")#"eb" menjadi "be"
#berlaku pada "eb, ec, ed, sampai ey"
#2. karakter sengau nga
.replace("01320061", "0193")#"Na" menjadi "nga"
.replace("01930069", "01FC")#"ngai" menjadi "ngi"
#berlaku nga sampai ngo
#dan seterusnya terurut mba-mpa
#3. rule karakter utama
.replace("00620061", "00C3")#ba
.replace("00630061", "00C4")#ca
.replace("00640061", "00C5")#da
.replace("00660061", "00C7")#fa
.replace("00670061", "00C8")#ga
.replace("00680061", "00C9")#ha
.replace("006A0061", "00CB")#ja
#dan seterusnya sampai karakter ya
```

Gambar 7. Impelementasi Rule Base pada sistem

Pada Gambar 7 merupakan proses penerapan *rule base* pada sistem, dimana proses ini terbagi menjadi 3 buah tahap. Tahap pertama proses pengenalan karakter vokal, seluruh karakter konsonan digabungkan dengan karakter vokal yang terdiri dari "i, u, e, o", selajutnya akan diproses untuk mendapatkan alamat dari hasil penggabungan karakter utama karakter vokal "a" dan karakter vokal yang digunakan "i/ u/ e/ o". Pada Gambar 7 karakter vokal "ba" akan menjadi "bai" yang selanjutnya akan menghasilkan alamat "012C" yang dimiliki oleh karakter "bi". Proses

ini dilakukan berulang dimulai dari karkater "ba" sampai dengan karakter "ya".

Proses kedua merupakan proses penerapan *rule base* pada karakter sengau. karakter sengau pada tahap ini diperlakukan hampir yang sama dengan karakter lain, dimana karkater sengau yang memiliki tambahan huruf vokal "a" akan dilakukan perhitungan dengan menambahkan karakter yang dituju. Pada Gambar 7 karakter sengau yang dijadikan sampel sesuai dengan permasalahan pada file uji adalah karakter "nga", hanya saja yang membedakan karakter tersebut tidak menggunakan huruf kecil namum menggunakan karakter "N", dimana karakter tersebut telah dijabarkan sebelumnya menjadi karakter "Na", dan mendapatkan pengalamatan 0132, yang berikutnya digabungkan degan karakter "a", sehingga menghasilkan alamat 0193 dengan alamat "nga". Pada baris berikutnya adalah karakter "ngi" yang merupakan hasil dari penggabungan yang berasalkan dari alamat karakter "nga" yang ditambahkan dengan karakter "i" sehingga menghasilkan aamat heksadesimal 01FC, atau alamat dari karakter "ngi".

Peroses ketiga adalah peroses *rule base* pada karakter utama, pada proses ini berjalan sebuah perhitungan dimana alamat heksadesimal dari konsonan akan digabungkan dengan alamat karakter vokal "a", yang selanjutnya dilakukan perhitungan manual, dimana hasil perhitungan tersebut digunakan sebagai *output replacement* dari masing masing karakter, hasil dari proses ini yang selanjutnya akan digunakan untuk memanggil karakter pada proses heksadesimal menjadi karakter.

```
#heksa ke huruf
d=(c
#karakter Utama
.replace("0061", "a")
.replace("00C3", "ba")
.replace("00C4", "ca")
.replace("00C5", "da")
.replace("00C7", "fa")
.replace("00C8", "ga")
.replace("00C9", "ha")
.replace("00CB", "ja")
#dan seterusnya sampai karakter ya
.replace("0069", "i")
.replace("012C", "bi")
#dan seterusnya sampai karakter ya
.replace("0075", "u")
.replace("0138", "bu")
#dan seterusnya sampai karakter ya
.replace("0065", "e")
.replace("0128", "be")
#dan seterusnya sampai karakter ya
.replace("006f", "o")
.replace("0132", "bo")
#dan seterusnya sampai karakter ya
#karakter sengau
.replace("019C", "mba")
#dan seterusnya sampai karakter ya
.replace("0193", "nga")
.replace("01FC", "ngi"))
```

Gambar 8. Heksadesimal menjadi Karakter

Gambar 8 merupakan proses perubahan dari alamat heksadesimal menjadi karakter latin, pada

tahap ini deret angka heksadesimal yang didapat akan dirubah menjadi karakter latin sehingga membentuk kalimat yang dijadikan sebagai *output* dari sistem.

```
)
#hasil
print (d)
makagorina, daaimawaana, ngarana, sabeya
```

Gambar 9. *Output* Pada sistem

Gambar 9 merupakan hasil dari proses transliterasi, dimana kalimat yang diinputkan dalam bentuk Aksara Bima digital dapat dikembalikan menjadi huruf latin. terdapat perbedaan *output* pada kata sengau "Makanggorina" dengan hasil proses transliterasi menjadi "makagorina", dan kata "Ndai Ma wa'ana" dimana sistem menampilkan karakter "daaimawaana". Permasalahan tersebut berasal dari persamaan karakter "ngga/nggi/nggu/ngge/nggo" yang akan disimbolkan dengan Aksara "ga/gi/gu/ge/go" dan karakter "Nda/i/u/e/o" yang akan disimbolkan dengan "da/di/du/de/do".

4. HASIL DAN PEMBAHASAN

Proses pengujian algoritma pada penelitian ini menggunakan data set yang terdapat pada buku [1]. skema ini dipilih untuk memastikan implementasi pada *rule base* telah diterapkan sesuai dengan aturan yang dimiliki oleh Aksara Bisma, adapun data yang terdapat pada buku *Aksara Bima Peradaban Lokal yang sempat hilang* adalah pada Tabel I-IX. dari tahap proses ini mendapatkan akurasi sebesar 90.64% dari 171 aturan yang diterapkan, aturan tersebut terdiri dari pengimplementasian dari aturan seperti, penjabaran aturan huruf dan simbol terdapat 51 aturan, bentuk karakter bima per huruf yang dikolaborasi dengan huruf vokal terdapat 85 aturan, bentuk sengau Aksara Bima 35 aturan, perlakuan terhadap huruf mati 28 aturan. Dengan model Perhitungan *error* sistem sebagaimana Persamaan (7).

$$\% Error = (16/171) * 100 = 9.36\% \quad (7)$$

Pada Persamaan (7) dapat dijelaskan terdapat ketidak sesuaian data validasi sebanyak 16 *case* 15 *case* berasal dari huruf sengau "Mba" sebanyak 5 *case*, "Nda" sebanyak 5 *case*, "Ngga" sebanyak 5 *case* pada karakter ai 1 *case* dengan total 16 *case* yang diujikan dengan 171 data uji dan dikalikandengan 100% sehingga mendapat error sistem 9.36%, sehingga didapat

akurasi 90.64%. Proses tidak dapat mencapai tingkat akurasi 100% dikarenakan pada *case* karakter sengau, sistem belum dapat membedakan inputan pada karakter "ia", "mba", "nda", dan "ngga" yang akan tetap dikenali menjadi karakter "ya" "ba", "da" dan "ga" dikarenakan atura baca yang dimiliki kater tersebut.

5. KESIMPULAN DAN SARAN

Kesimpulan yang dihasilkan dari penelitian ini adalah algoritma *string replacement* yang dioptimalkan dengan metode *string replacement* dapat melakukan proses transliterasi Aksara Bima menjadi huruf latin dengan akurasi 90.64% dan dapat disimpulkan pula pemanfaatan bilangan heksadesimal pada proses *string replacement* dapat dikolaborasi guna pada proses penerapan *rule base* atau aturan baku penulisan Aksara Bima.

Saran Pertama untuk menyelesaikan permasalahan yang dihadapi pada karakter sengau adalah menambahkan proses penambahan kata bahasa Bima untuk membuat proses pencocokan string sehingga hasil pengujian akan melewati satu buah proses pencocokan kalimat bahasa Bima untuk dapat memproses karakter segau secara keseluruhan. Saran kedua pada penelitian ini belum terimplementasikannya penggunaan tanda baca awalan kata titik koma spasi dalam paragraf sehingga diperlukan pengkajian mendalam terkait data sampel pada topik ini. Saran ketiga yang dapat disampaikan penulis pada proses transliterasi Aksara Bima menggunakan algoritma *string replacement* ini baru bersifat implementasi metode dan dapat dirancang sebuah aplikasi berbasis *smart phone* untuk dijadikan aplikasi yang dapat dijalankan dan diujikan kepada masyarakat untuk keperluan dunia pendidikan, pelestarian budaya, penelitian, dan bisnis.

UCAPAN TERIMA KASIH

Ucapan terimakasih kami ucapkan kepada se seluruh team Anggota Hj. Siti Mariam karena dengan adanya buku tersebut dapat dilakukanya proses validasi hasil uji pada penelitian ini.

DAFTAR PUSTAKA

- [1] R. S. Maryam Siti, M. Sulaiman and S. Abubakar *Aksara Bima Peradaban Lokal yang sempat hilang*. Samparaja, Bima: Alam Tara institute, Mataram, 2013. pp.1-64.
- [2] E. Alfian, "Penggunaan Unsur Aksara Nusantara Pada Huruf Modern," *Ultim. J. Komun. Vis.*, vol. 7, no. 1, pp. 42–48, 2016, doi: 10.31937/ultimart. v7i1.370.

- [3] Governor, *Bali Governor Regulation number 80 of 2018 About Protection and Use of Bali, Aksara, And Literature as well as the Implementation of the Bali language*. 2018, pp. 1–9.
- [4] Kementerian Komunikasi dan Informatika “Journal Teknologi Informasi dan Komunikasi,” *J. Teknol. Inf. Dan Komun.*, vol. 5, no. 1, pp. i–viii, 2016.
- [5] A. Aranta, Ig. A. Gunadi, and G. Indrawan, “Utilization of Hexadecimal Numbers in Optimization of Balinese Transliteration String Replacement Method,” *2018 Int. Conf. Comput. Eng. Netw. Intell. Multimedia, CENIM 2018 - Proceeding*, pp. 131–136, 2018, doi: 10.1109/CENIM.2018.8711118.
- [6] I.G. Andika, C.P. Yanti. and G. indrawan. “Algorithm to Avoid Overlapping Vowel Signs in Latin to Balinese Script Transliteration Method,” *Int. Conf. New Media Stud.*, pp. 48-53., 2019, doi: 10.1109/CONMEDIA46929.2019.8981858.
- [7] V. Atina, S. Palgunadi, and W. Widiarto, “Program Transliterasi Antara Aksara Latin dan Aksara Jawa dengan Metode FSA,” *J. Teknol. Inf. ITS smart*, vol. 1, no. 2, p. 60, 2012, doi: 10.20961/its.v1i2.592.
- [8] I. P. A. E. D. Udayana, M. Sudarma, and I. N. S. Kumara, “Balinese Latin Text Becomes Aksara Bali Using Rule Base Method,” *Int. J. Res. IT, Manag. Eng.*, vol. 07, no. 05, pp. 1–7, 2017.
- [9] R. Yulianti, I. G. P. S. Wijaya, and F. Bimantoro, “Pengenalan Pola Tulisan Tangan Suku Kata Aksara Sasak Menggunakan Metode Moment Invariant dan Support Vector Machine,” *J. Comput. Sci. Informatics Eng.*, vol. 3, no. 2, pp. 91–98, 2019, doi: 10.29303/jcosine.v3i2.181.
- [10] E. D. J. Utari Ms, I. G. P. S. Wijaya, and F. Bimantoro, “Pengenalan Pola Tulisan Tangan Suku Kata Aksara Sasak Menggunakan Metode Integral Projection dan Neural Network,” *J. Comput. Sci. Informatics Eng.*, vol. 3, no. 1, p. 19, 2019, doi: 10.29303/jcosine.v3i1.222.
- [11] I. B. K. Widiartha, and F. Soesianto, “Pembuatan perangkat lunak transliterator Aksara latin ke Aksara Bali dan Aksara Bali ke Aksara latin,” Gajah Mada University, 2001.
- [12] I. G. Andika, D. E. Saputra, C. P. Yanti, and G. Indrawan. “Mengenal Aksara Bali Balinese Script game education based on mobile application,” vol. 04, no. 01, pp. 1–6, 2019.
- [13] G. Indrawan and I. K. Pramarta, “Pengembangan Aplikasi Mobile Pembelajaran Transliterasi Tulisan Latin ke Aksara Bali,” *J. Pekommas*, vol. 4, no. 2, p. 123, 2019, doi: 10.30818/jpkm.2019.2040202.
- [14] G. P, Hoetomo, M. Ulum, and Haryanto. “Transliterasi Huruf Latin Bahasa Indonesia ke Aksara Jawa Menggunakan Metode Snakecut,” *fortei7*, vol. 2, no. 1, pp. 96–101, 2019.
- [15] A. Syauqi and A. N. Hidayah, “Implementasi Komputasi Paralel untuk Optimalisasi Komputasi Pada Aplikasi Transliterasi Huruf Latin ke Aksara Jawa,” *J. Online Inform.*, vol. 3, no. 1, p. 29, 2018, doi: 10.15575/join.v3i1.179.
- [16] Unicode, “Balinese Character Unicode,” *Unicode, Inc.*, 2020. [Online]. Available: <https://unicode.org/charts/PDF/U1B00.pdf>.