

Rancang Bangun Aplikasi Transliterasi Aksara Bali Menjadi Huruf Latin Menggunakan Metode Rule Based Pada UTF-16 Berbasis Android

I Gede Ketut Saditha Putra¹, Arik Aranta¹, I Gede Pasek Suta Wijaya¹, I Gede Andika²

¹ Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram Jl. Majapahit No.62, Gomong, Kec. Selaparang, Kota Mataram, Nusa Tenggara Barat, Indonesia

² Program Studi Sistem Komputer, Fakultas Teknologi dan Informatika, Institut Bisnis dan Teknologi Indonesia

Jl. Tukad Pakerisan No.97, Panjer, Denpasar Selatan, Kota Denpasar, Bali, Indonesia

e-mail: [arikaranta, gpsutawijaya]@unram.ac.id,
putrasaditha@gmail.com, gdandika@instiki.ac.id

WhatsApp contact: +6287865668341¹, +625785576822¹, +629685410790²

Received : August, 2023	Accepted : December, 2023	Published : December, 2023
-------------------------	---------------------------	----------------------------

Abstract

Balinese script is a script that has been used by Balinese tribes since ancient times, as evidenced by the discovery of 2,103 ancient lontars using Balinese script in Klungkung, Bali. Currently, Google already has a virtual keyboard that can be downloaded to a smartphone. This Google keyboard already has Balinese script mode that the letters was based on Unicode, but it still doesn't have a feature that can read Balinese script into Latin letters. By combining these two problems, a study was designed about making applications that can preserve Balinese script by utilizing the Balinese script keyboard feature provided by Google to be transliterated into Latin letters. In this study, a rule based method is used to perform transliteration in order to be able to adjust to the rules of reading Balinese script correctly. It is also used to provide ID for each Balinese character using hexadecimal based on the Unicode Transformation Format-16 code for each character. The application was built by implementing of total 940 rules and 33.253 words, after testing the application the result obtained from the transliteration reach 99.4% of succeed from 1104 Balinese script words.

Keywords: Balinese script, transliteration, rule based, Unicode, Android

Abstrak

Aksara Bali merupakan aksara yang telah digunakan oleh suku Bali sejak zaman dahulu, terbukti dengan ditemukannya 2.103 lontar kuno yang menggunakan aksara Bali di Klungkung, Bali. Saat ini Google sudah memiliki keyboard virtual yang bisa diunduh ke smartphone. Keyboard Google ini sudah memiliki mode Aksara Bali yang hurufnya berbasis Unicode, namun masih belum memiliki fitur yang dapat membaca Aksara Bali menjadi huruf Latin. Dengan menggabungkan kedua permasalahan tersebut, maka dirancanglah sebuah penelitian tentang pembuatan aplikasi yang dapat melestarikan aksara Bali dengan memanfaatkan fitur keyboard aksara Bali yang disediakan oleh Google untuk ditransliterasikan menjadi huruf latin. Dalam penelitian ini digunakan metode rule based untuk melakukan transliterasi agar dapat menyesuaikan dengan kaidah pembacaan aksara Bali dengan benar. Digunakan juga untuk memberikan ID setiap karakter Bali menggunakan heksadesimal berdasarkan kode Unicode Transformation Format-16

untuk setiap karakter. Aplikasi ini dibangun dengan mengimplementasikan total 940 aturan dan 33.253 kata, setelah dilakukan pengujian aplikasi hasil transliterasi yang diperoleh mencapai 99.4% keberhasilan dari 1104 kata aksara Bali.

Kata Kunci: *Balinese script, transliteration, rule based, Unicode, Android*

1. INTRODUCTION

Bali merupakan pulau Dewata yang memiliki beragam kesenian dan budaya, salah satu budaya yang dimilikinya adalah aksara Bali. Di Bali, aksara Bali sudah menjadi bahan pembelajaran bagi siswa-siswa agar nantinya aksara ini tetap dapat dilestarikan. Namun saat ini, penggunaan aksara Bali semakin berkurang dikarenakan banyaknya penggunaan huruf latin [1].

Aksara Bali merupakan aksara yang sudah digunakan oleh suku Bali sejak jaman dahulu, terbukti dengan ditemukannya sebanyak 2.103 lontar kuno yang menggunakan aksara Bali di Klungkung, Bali. Selain lontar kuno yang berasal dari Klungkung terdapat juga lontar-lontar lainnya dan ada pula yang berasal dari kerajaan-kerajaan yang pernah menduduki Bali [2]. Oleh karena itu, kelestarian aksara Bali perlu dijaga selain sebagai media komunikasi juga sebagai cara agar mampu mengetahui sejarah yang tercatat pada lontar-lontar kuno Bali. Dalam upaya melestarikan aksara Bali tersebut, sudah banyak penelitian yang berkaitan dengan Aksara Bali. Khususnya penelitian terkait membuat suatu sistem yang mampu menerjemahkan huruf latin yang biasa kita gunakan menjadi aksara Bali. Misalnya pada penelitian terdahulu yang melakukan pembuatan aplikasi berbasis Android yang mampu menerjemahkan huruf latin menjadi Aksara Bali, aplikasi yang berhasil dibuat mampu menghasilkan ketepatan sebesar 92,72% dari sebanyak 3915 *dataset* [3].

Selain itu, terdapat juga penelitian yang membuat program yang dapat mengubah huruf latin menjadi Aksara Bali dengan ketepatan sebesar 90,67% [4]. Berdasarkan permasalahan tersebut, dilakukan penelitian yang bertujuan merancang aplikasi berbasis Android yang mampu men-transliterasi dari aksara Bali menjadi huruf latin. Penelitian ini menggunakan metode *rule based* sebagai metode untuk melakukan transliterasi. Selain metode *rule based*, diterapkan juga optimalisasi metode *string replacement* terhadap heksadesimal UTF-

16(16-bit *Unicode* Transformation Format) dari aksara Bali, nantinya setiap input-an yang terdapat pada keyboard akan diubah menjadi heksadesimal sesuai dengan konversinya berdasarkan *Unicode* versi UTF-16 yang selanjutnya akan digunakan sebagai ID untuk proses *rule based* hingga menjadi transliterasi yang sesuai. Digunakan UTF-16 dalam pengimplementasian *Unicode* karena pengkodean UTF-16 berbasis dari bilangan heksadesimal, dimana bilangan heksadesimal yang dihasilkan ini memiliki jumlah digit yang tetap dalam aksara Bali, yaitu sebanyak empat digit sehingga nantinya dalam pemrosesan sistem dalam mengurangi kekeliruan dalam pemberian ID. Aplikasi yang dibuat akan memanfaatkan *GBoard Google keyboard* untuk proses pemberian input Aksara Bali. Dengan transliterasi yang menggunakan data dari UTF-16 ini, aplikasi nantinya akan mampu melakukan transliterasi terhadap karakter digital Aksara Bali yang bersumber dari segala sumber yang menggunakan *Unicode*. Selain itu, pada aplikasi ini akan menggunakan kamus aksara Bali untuk penyesuaian transliterasi terhadap Bahasa Bali agar hasil transliterasi yang dilakukan dapat lebih akurat. Diharapkan dengan menerapkan kedua metode tersebut dapat dihasilkan aplikasi berbasis Android yang mampu men-transliterasi Aksara Bali dengan baik khususnya terhadap teks-teks yang menggunakan Bahasa Bali.

2. Tinjauan Pustaka

Untuk menunjang terlaksananya penelitian ini, didapat beberapa penelitian yang memiliki konsep yang serupa dengan penelitian ini untuk dijadikan referensi sebagai berikut. Penelitian pertama memiliki hasil akhirnya berupa aplikasi yang mampu melakukan transliterasi dari huruf latin menjadi Aksara Bali. Dalam proses transliterasi, penelitian ini menggunakan bilangan heksadesimal sebagai id atau kode untuk setiap karakter input-an agar optimalisasi transliterasi dapat ditingkatkan. Untuk akurasi dari aplikasi yang dibuat, aplikasi ini

menghasilkan akurasi sebesar 92,72% dari total 3915 data tes [3]. Terdapat juga penelitian yang hampir serupa yang letak perbedaannya pada penelitian yang ini tidak diimplementasikan pada Android dan tidak dilakukan optimalisasi menggunakan heksadesimal. Untuk metode transliterasinya juga menggunakan rule based dan target transliterasinya dari huruf latin menjadi Aksara Bali. Penelitian ini mendapatkan akurasi sebesar 90,67% dengan data uji yang digunakan bersumber dari lontar [4].

Terdapat juga penelitian dengan hasil akhir berupa program yang mampu melakukan transliterasi secara dua arah dari huruf latin menjadi Aksara Bali maupun sebaliknya. Penelitian ini berhasil membangun sistem yang dapat menyesuaikan dengan aturan-aturan dari setiap karakter Aksara Bali [5]. Selain penelitian transliterasi pada Aksara Bali, terdapat juga beberapa penelitian transliterasi terhadap Aksara yang lain. Misalnya penelitian pembuatan sistem yang dapat melakukan transliterasi dari Aksara Bali menjadi huruf latin. Dalam proses transliterasinya penelitian ini menggunakan metode *rule based* serta pendekatan heksadesimal untuk meningkatkan optimalisasi transliterasi. Akurasi dari sistem yang dibuat pada penelitian ini sebesar 90,64% dari 171 aturan yang diterapkan pada sistem [6]. Terdapat juga penelitian transliterasi Aksara Jawa yang proses transliterasinya melibatkan image processing dan menggunakan metode Snakecut. Dengan input yang berupa gambar akan diproses menggunakan aplikasi Microsoft Visual Studio dengan library EmguCV. Sedangkan untuk akurasinya, penelitian ini mendapatkan akurasi sebesar 85% dari 20 data uji [7]. Pada penelitian lainnya ada juga yang melakukan transliterasi Aksara Jawa namun memanfaatkan *Finite State Automata*. Penelitian ini menghasilkan sistem yang dapat menerjemahkan Bahasa Jawa menjadi Aksara Jawa dan sebaliknya. Sistem ini menggunakan finite state automata dalam proses penerjemahannya. Untuk pengujian dari sistem, dilakukan pengujian oleh pakar Bahasa Jawa dengan hasil 92% keberhasilan terjemahan dari Bahasa Jawa ke Aksara Jawa dan 93,8% keberhasilan terjemahan dari Aksara Jawa ke Bahasa Jawa [8]. Terdapat juga penelitian lainnya yang membuat aplikasi transliterasi tulisan latin ke Aksara Bali dengan menggunakan 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 [9]. Masih ada lagi penelitian terhadap Aksara Jawa, aplikasi yang dibangun yang mampu melakukan transliterasi dari huruf latin menjadi Aksara Jawa. Penelitian ini bertujuan untuk mempersingkat proses transliterasi dari aplikasi hasil penelitian yang sudah ada sebelumnya. Untuk mempersingkat proses transliterasi digunakan multithread agar dapat melakukan komputasi secara parallel. Aplikasi ini mampu mengurangi waktu komputasi sebesar 97,05% dari waktu komputasi rata-rata [10].

2.1 Dasar Teori

Teori-teori dasar atau umum yang digunakan dalam penelitian ini yaitu sebagai berikut.

2.2.1 Aksara Bali

Aksara merupakan salah satu jenis simbol visual dari suatu bahasa. Bahasa Bali dapat ditulis dengan dua jenis simbol yaitu dengan Aksara Bali dan tulisan Bali Latin. Menurut sejarah Aksara Bali, Aksara Bali diambil dari huruf Jawa, sedangkan huruf Jawa mengambil dari huruf *pallawa/dewanegari*, hanya bentuknya secara evolusi sudah berubah sehingga menjadi aksara yang sering dipakai saat ini [11]. Berdasarkan atas bentuk dan fungsinya, Aksara Bali dibagi atas dua jenis yakni aksara biasa dan aksara suci. Aksara biasa ini terdiri atas aksara Wreastra dan Swalalita. Disebut aksara biasa karena telah terbiasa dipergunakan oleh masyarakat Bali di dalam tulis-menulis untuk memenuhi kebutuhan hidupnya sehari-hari dalam berhubungan dengan sesama melalui aksara. Sedangkan aksara suci terbagi atas dua kelompok, yakni aksara *Wijaksana* dan *Mondre*. Aksara *Wijaksana* pada hakikatnya adalah aksara Swalalita/Wreastra yang diberi pengangge atau busana aksara. Sedangkan aksara *Modre*, lebih tepat disebut lukisan, karena merupakan bentuk dari aksara *Swalalita* ditambah dengan beberapa pengangge aksara serta lukisan, sehingga menjadi "aksara mati", sulit untuk membacanya [12].

2.2.2 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware, dan aplikasi.

Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008[13].

2.2.3 Rule Based

Sistem pakar adalah program komputer yang dirancang untuk memecahkan masalah dalam domain tertentu dengan melibatkan kepakaran atau keahlian manusia. Pengetahuan yang dibangun di dalam sistem biasanya diperoleh dari para ahli di lapangan. Sistem pakar dapat meniru proses berpikir dari para ahli dan membuat kesimpulan logis sesuai pengetahuan. Salah satu tipe sistem pakar yang populer adalah sistem berbasis aturan (rule based system). Pendekatan sistem berbasis aturan banyak dilakukan dalam implementasi sistem pakar karena memiliki beberapa keuntungan sebagai berikut[14]:

- a) Bersifat modular sehingga memudahkan pengembangan sistem karena untuk melakukan penambahan aturan tidak mengganggu struktur aturan yang sudah ada. Dalam sistem berbasis aturan, aturan dan program merupakan hal yang terpisah.
- b) Alami dan mirip dengan proses kognitif manusia sehingga representasi sederhana aturan tersebut dalam bentuk IF-THEN mudah dipahami.

2.2.3 Heksa Desimal

Bilangan heksadesimal, sering disingkat dengan hex, adalah bilangan dengan basis 16, dan mempunyai 16 simbol yang berbeda, yaitu 0 sampai dengan 15[15]. Angka heksadesimal, sering disingkat dengan hex, adalah bilangan dengan basis 16, dan mempunyai 16 simbol yang berbeda, yaitu 0 sampai dengan 15. 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 terdapatnya karakter Aksara Bali pada Unicode membuat proses penelitian ini melakukan proses adopsi karakter unicode yang berdasarkan 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 meningkatkan penulisan bilangan angka dalam jumlah bilangan ribuan pada bilangan desimal. Bilangan desimal dengan angka 6015 dalam heksadesimal akan menjadi 177F[6].

2.2.4 String Replacement

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

```
1. input = "Belajar Aksara Bali itu sulit"  
2. a=(input  
3. .replace("sulit","sangat mudah"))  
4. print(a)
```

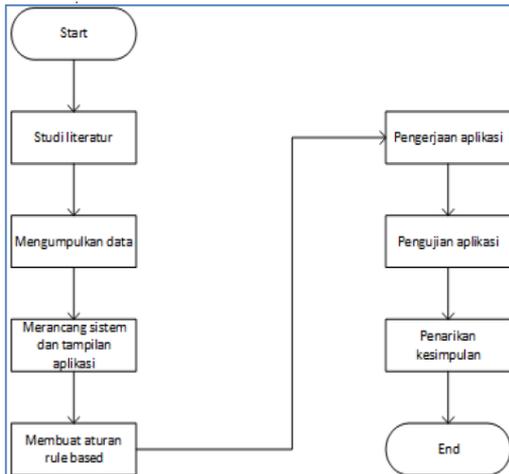
Belajar Aksara Bali itu sangat mudah

Gambar 1. Contoh *String replacement*

Pada Kode Sumber 2.1 merupakan penerapan sebuah proses string replacement. Pada proses tersebut terjadi perubahan pada karakter yang di input-kan dengan karakter yang dikasihkan, dimana karakter dengan masukan "Belajar Aksara Bali itu sulit" dan terjadi replacement kata "sulit", menjadi kata "sangat mudah" maka output yang dihasilkan adalah "Belajar Aksara Bali itu sangat mudah" [6].

3. Metode Penelitian

Dalam penelitian ini, terdapat tahap-tahap kegiatan yang dilakukan pada alur penelitian sebagai berikut. Seperti yang terlihat pada Gambar 2, tahapan penelitian ini terdiri dari tujuh tahap dimulai dari studi literatur hingga penarikan kesimpulan.



Gambar 2. Diagram alur penelitian.

3.1 Studi Literatur

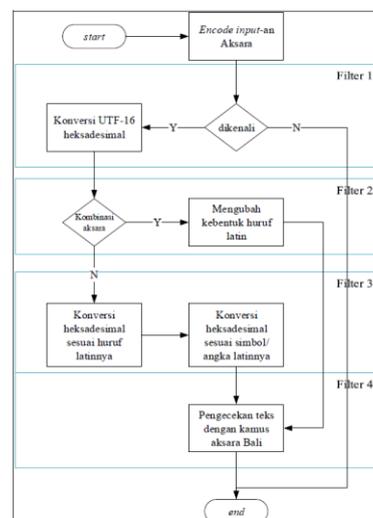
Pada tahap ini dilakukan studi literatur terhadap buku, jurnal, ataupun artikel yang berhubungan dengan aksara Bali maupun proses transliterasi untuk membantu dalam pengerjaan sistem. Kemudian hasil informasi yang telah didapat akan dikumpulkan dan dijadikan acuan pengerjaan sistem mulai dari metode hingga aturan dari perubahan aksara Bali. Dalam penelitian ini, data yang digunakan untuk membangun aturan transliterasi aksara Bali bersumber dari Kamus “Bali – Indonesia beaksara Latin dan Bali” [16], babadbali[17], dan beberapa artikel terkait aksara Bali.

3.2 Mengumpulkan Data

Untuk tahap pengumpulan data didapat bentuk digital dari aksara Bali yang telah terdata pada Unicode. Pada Unicode, Balinese “1B00-1B7F”. Data dari karakter aksara Bali dapat dilihat pada Tabel I. Tabel I. Data karakter aksara Bali Unicode karakter aksara Bali ini terdata pada blok. Sedangkan untuk dataset kata Bahasa Bali dan Indonesia didapat sebanyak 33.253 kata yang bersumber dari kamus Bahasa Bali dan kamus Bahasa Indonesia dimana kata-kata ini digunakan untuk menjadi kamus pemberian spasi dari hasil transliterasi setelah menjadi huruf latin. Kemudian untuk menjadi data uji ketepatan proses transliterasi didata sebanyak 1104 kata yang bersumber dari Kamus Bali-Indonesia dan beberapa buku belajar aksara Bali.

U+	1B0	1B1	1B2	1B3	1B4	1B5	1B6	1B7
0	ᬀ	ᬁ	ᬂ	ᬃ	ᬄ	ᬅ	ᬆ	ᬇ
1	ᬈ	ᬉ	ᬊ	ᬋ	ᬌ	ᬍ	ᬎ	ᬏ
2	ᬑ	ᬒ	ᬓ	ᬔ	ᬕ	ᬖ	ᬗ	ᬘ
3	ᬙ	ᬚ	ᬛ	ᬜ	ᬝ	ᬞ	ᬟ	ᬠ
4	ᬡ	ᬢ	ᬣ	ᬤ	ᬥ	ᬦ	ᬧ	ᬨ
5	ᬩ	ᬪ	ᬫ	ᬬ	ᬭ	ᬮ	ᬯ	ᬰ
6	ᬱ	ᬲ	ᬳ	᬴	ᬵ	ᬶ	ᬷ	ᬸ
7	ᬹ	ᬺ	ᬻ	ᬼ	ᬽ	ᬾ	ᬿ	ᬿ
8	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ
9	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ
A	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ
B	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ
C	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ
D	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ
E	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ
F	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ	ᬿ

3.3 Merancang Sistem dan Tampilan



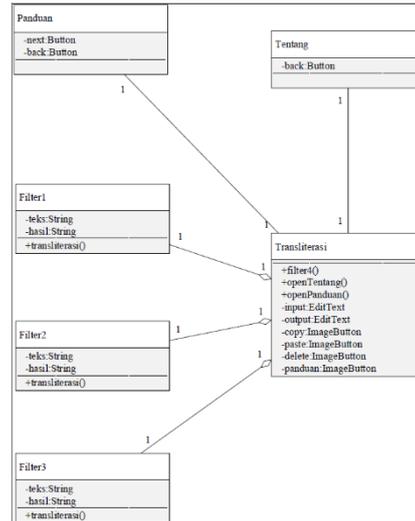
Gambar 3. Diagram alur proses transliterasi.

Dalam membuat rancangan sistem transliterasi, dibuat sebuah diagram alur untuk menggambarkan proses yang akan dilakukan

sistem untuk melakukan transliterasi pada Gambar 3. Sesuai diagram alur transliterasi, proses transliterasi didalam sistem akan terbagi menjadi empat tahap dimulai dari filter 1 hingga filter 4. Setiap filter menerima bentuk teks masing masing dan akan menghasilkan teks masing masing sesuai dengan tujuan mereka sebagai berikut:

- a) Filter 1, filter ini akan menerima bentuk awal dari teks aksara Bali. Filter ini akan berisikan rule based untuk mengubah teks aksara Bali menjadi kebentuk heksadesimalnya sesuai pengkodean dalam Unicode.
- b) Filter 2, filter ini menerima hasil pengkodean dari filter 1. Pada filter ini, rule based yang diberikan berdasarkan aturan pembacaan aksara Bali karena pada filter ini bertujuan untuk mencari karakter aksara Bali yang merupakan kombinasi dari beberapa aksara untuk kemudian aksara ini akan di transliterasikan kebentuk huruf latinnya.
- c) Filter 3, filter ini menerima hasil teks yang telah dilalui dari filter 2. Sesuai yang dijelaskan pada filter 2 tadi, diketahui hanya teks aksara kombinasi yang diubah kebentuk latinnya sedangkan aksara yang berdiri tunggal masih berupa kode heksadesimal. Aksara yang berdiri tunggal ini merupakan tujuan dari filter 3, dimana pada filter ini teks tersebut akan diubah kebentuk latinnya sehingga hasil teks akan menjadi huruf latin secara utuh.
- d) Filter 4 merupakan filter terakhir dimana teks yang masuk pada filter ini sudah berupa teks murni latin. Tujuan dari filter ini adalah pemberian spasi, tradisionalnya aksara Bali tidak memiliki spasi sehingga ketika diubah kebentuk latin teks akan tidak memiliki spasi sama skali. Filter ini memberikan spasi berdasarkan kamus kata yang didata dari kamus Bahasa Bali dan Indonesia. Ketika teks memiliki kata yang terdata pada kamus maka tugas filter ini akan memberikan spasi dengan kata selanjutnya.

Kemudian untuk gambaran implementasi sistem dalam Android, dibuat sebuah kelas diagram yang berisikan rancangan implementasi *class-class* dari sistem pada Gambar 4.



Gambar 4. Class diagram aplikasi.

3.4 Membuat Aturan Rule Based

Pada tahap transliterasi aksara Bali, sistem akan diimplementasikan beberapa *rule based* berdasarkan aturan pembacaan aksara Bali agar menghasilkan teks latin yang sesuai. Rule Based tersebut dikategorikan menjadi empat filter, mulai dari filter 1 hingga 4. Berikut merupakan pembuatan rule based yang diimplementasikan pada sistem.

3.4.1 Filter 1

Tabel II. Sample *rule based filter 1*.

No.	Aksara	Encode	Pengkodean Unicode
1	ᮀ	%E1%AC%80	1B00
2	ᮁ	%E1%AC%81	1B01
3	ᮂ	%E1%AC%82	1B02
4	ᮃ	%E1%AC%83	1B03
5	ᮄ	%E1%AC%84	1B04
6	ᮅ	%E1%AC%85	1B05
7	ᮆ	%E1%AC%86	1B06
8	ᮇ	%E1%AC%87	1B07

Filter 1 berisikan rule based untuk perubahan bentuk aksara Bali menjadi pengkodean Unicode. Sample rule based yang diimplementasikan dapat dilihat pada Tabel II.

Dari Tabel terlihat bentuk aksara bali yang diberikan rule based adalah yang berbentuk encode. Sedangkan sample implementasi filter ini didalam sistem dapat dilihat pada Gambar 5.

```

1. //filter 1
2. String encde = URLEncoder.encode(text, "UTF-8");
3. text=encde
4. .replace("%E1%AC%80", "1B00")//ulu ricem
5. .replace("%E1%AC%81", "1B01")//ulu candra
6. .replace("%E1%AC%82", "1B02")//cecek
7. .replace("%E1%AC%83", "1B03")//surang
8. .replace("%E1%AC%84", "1B04")//bisah
9. .replace("%E1%AC%85", "1B05")//akara
10. .replace("%E1%AC%86", "1B06")//akara tedung
11. .replace("%E1%AC%87", "1B07")//ikara

```

Gambar 5. Sample rule based filter 1.

Dapat dilihat pada line 2 teks aksara Bali diubah kebentuk encode kemudian mulai dari line 3 diberikan method String replacement sebagai sarana untuk menerapkan rule based yang ada.

3.4.2 Filter 2

Filter 2 merupakan filter yang berisikan rule based paling banyak karena pada filter ini mengatur kombinasi-kombinasi aksara yang ada di aksara Bali. Sample rule based yang diimplementasikan dapat dilihat pada Tabel III.

Tabel III. Sample rule based filter 2.

No.	Aksara	Filter 1	Filter 2
1	ᮊᮧ	1B331B44	h
2	ᮊᮦ	1B331B3E	he
3	ᮊᮥ	1B331B35	ha
4	ᮊᮤ	1B331B38	hu
5	ᮊᮣ	1B331B36	hi
6	ᮊᮢ	1B331B3F	hai
7	ᮊᮡ	1B331B02	hang
8	ᮊᮠ	1B331B42	he
9	ᮊᮟ	1B331B03	har
10	ᮊᮞ	1B331B3A	hre
11	ᮊᮝ	1B331B04	hah
12	ᮊᮜ	1B331B3C	hle
13	ᮊᮛ	1B331B40	ho

Dapat dilihat pada Tabel III beberapa sample rule based terhadap kombinasi aksara berkode 1B33(ha). Berdasarkan sample table, diimplementasikan rule ini kedalam sistem yang dapat dilihat pada sample di Gambar 6.

```

1. String f2=text;
2. text = f2
3. .replace("1B331B44", "h") //ha
4. .replace("1B331B3E", "he")
5. .replace("1B331B35", "ha")
6. .replace("1B331B38", "hu")
7. .replace("1B331B36", "hi")
8. .replace("1B331B3F", "hai")
9. .replace("1B331B02", "hang")
10. .replace("1B331B42", "he")
11. .replace("1B331B03", "har")
12. .replace("1B331B3A", "hre")
13. .replace("1B331B04", "hah")
14. .replace("1B331B3C", "hle")
15. .replace("1B331B40", "ho")

```

Gambar 6. Sample rule based filter 2.

Dapat dilihat pada line 3 mulai diberikan String replacement untuk pemberian rule terhadap kombinasi aksara yang ada. Sebagai contoh pada line 3 terdapat kode 1B33 yang merupakan kode untuk aksara “ha” berdampingan dengan kode 1B44 yang merupakan aksara adeg-adeg(menghapus “a”), maka pada String replacement kedua kode berdampingan ini diubah menjadi pembacaan latin mereka yaitu “h” begitu seterusnya. Rule ini diberikan untuk seluruh kombinasi aksara Bali yang ada sehingga jumlah rule filter ini sebanyak 780 rule.

3.4.3 Filter 3

Filter 3 hampir mirip dengan filter 1 dimana pada filter 1 teks aksara Bali diubah menjadi kode Unicode sedangkan pada filter 3 kode Unicode diubah menjadi latin. Pada filter ini tersisa beberapa kode Unicode yang belum di transliterasi menjadi latin di filter 2 karena kode Unicode yang tersisa merupakan karakter aksara yang tidak merupakan kombinasi. Beberapa sample rule based untuk filter ini dapat dilihat pada Tabel IV.

Tabel IV. Sample rule based filter 3.

No.	Aksara	Pengkodean Unicode	Latin
1	ᮍ	1B00	m
2	ᮎ	1B01	ng
3	ᮏ	1B02	ng
4	ᮐ	1B03	r
5	ᮑ	1B04	h
6	ᮒ	1B05	a
7	ᮓ	1B06	a
8	ᮔ	1B07	i

Sedangkan untuk implementasinya dalam sistem masih sama seperti filter sebelumnya, yaitu menggunakan String replacement. Beberapa sample implementasi rule dapat dilihat pada Gambar 7.

```

1. String f3=text;
2. text = f3
3. .replace("1B00","m")
4. .replace("1B01","ng")
5. .replace("1B02","ng")
6. .replace("1B03","r")
7. .replace("1B04","h")
8. .replace("1B05","a")
9. .replace("1B06","a")
10. .replace("1B07","i")
11. .replace("1B08","i")
12. .replace("1B09","u")
13. .replace("1B0A","u")
14. .replace("1B0B","re")
15. .replace("1B0C","ro")
16. .replace("1B0D","le")
17. .replace("1B0E","lo")

```

Gambar 7. Sample rule based filter 3.

Sama seperti sebelumnya, String replacement dimulai dari line 3 untuk pemberian rule based dimana pada filter ini rule diberikan untuk mengubah sisa sisa kode Unicode menjadi kebentuk latin mereka masing-masing.

3.4.4 Filter 4

Filter 4 merupakan filter terakhir dari proses transliterasi. Berbeda dengan filter sebelumnya, filter ini menerima teks yang sudah murni berbentuk latin. Tujuan dari filter ini adalah untuk pemberian spasi dari teks latin yang telah ditransliterasikan. Untuk pemberian spasinya, filter ini diberikan total sebanyak 33.253 kata sebagai patokan pengenalan kata yang diberikan spasi. Digunakan dinamik programming untuk pencocokan teks dengan kamus data yang ada, dimana kata yang cocok dengan panjang karakter terbanyak yang menjadi target pemberian spasi.

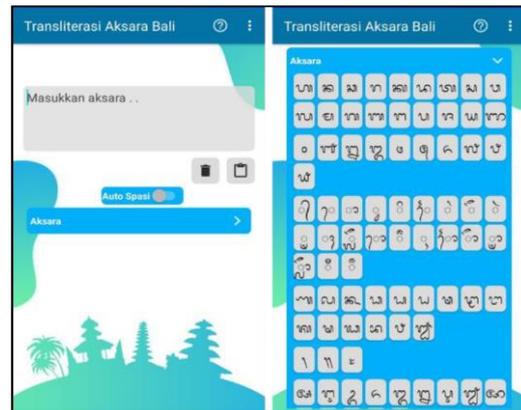
4. HASIL DAN PEMBAHASAN

Berdasarkan pengerjaan sistem yang telah dilakukan, didapat pembahasan yang bisa dijelaskan sebagai berikut.

4.1 Tampilan Aplikasi

Dari aplikasi yang telah dibuat, aplikasi memiliki empat buah halaman, yaitu halaman transliterasi, panduan, contoh, dan tentang. Ketika pertama kali membuka aplikasi, pengguna akan langsung ditujukan ke halaman

transliterasi. Halaman transliterasi dapat dilihat pada Gambar 8.



Gambar 8. Tampilan halaman transliterasi.

Pada halaman transliterasi ini, pengguna dapat memasukkan teks aksara Bali untuk ditransliterasikan. Selain itu, pengguna juga diberikan beberapa tombol untuk memudahkan mengoperasikan aplikasi, mulai dari tombol temple, hapus, cetak, karakter aksara, auto spasi, dan menu untuk mengakses halaman lain yang tersedia pada aplikasi. Kemudian terdapat halaman panduan, dimana pada halaman ini pengguna akan dipandu dengan gambar dan teks bagaimana cara menggunakan aplikasi.



Gambar 9. Tampilan halaman panduan.

Selanjutnya terdapat halaman contoh, pada halaman ini pengguna dapat menyalin teks aksara Bali sebagai sample untuk melakukan transliterasi ataupun untuk digunakan pada aplikasi lainnya.



Gambar 10. Tampilan halaman contoh.

Dan terakhir terdapat halaman tentang yang memuat deskripsi singkat tentang aplikasi Transliterasi Aksara Bali.



Gambar 11. Tampilan halaman tentang

4.2 Pengujian Aplikasi

Untuk menentukan apakah aplikasi telah berjalan sesuai dengan yang diharapkan, maka dilakukan pengujian. Pengujian yang dilakukan ada dua, yaitu pengujian akurasi dan pengujian *blackbox*.

4.2.1 Pengujian Akurasi

Pengujian akurasi digunakan untuk menentukan seberapa akurat transliterasi yang dihasilkan oleh aplikasi ini. Pengujian ini menggunakan data kata dan kalimat aksara Bali yang bersumber dari Kamus Bali-

Indonesia[16], babadbali[17], beberapa nama jalan di Bali, dan beberapa artikel terkait aksara Bali. Dengan total data uji sebanyak 1104 kata, aplikasi berhasil melakukan transliterasi yang sesuai sebanyak 1096 dan transliterasi yang tidak sesuai sebanyak 8 kata. Sehingga akurasi dari aplikasi dapat diketahui dengan persamaan (1).

$$\text{Akurasi} = (1104 - 8) / 1104 \times 100 = 99.4\% \quad (1)$$

Dari pengujian yang telah dilakukan terdapat beberapa hasil transliterasi yang tidak sesuai, hasil transliterasi tersebut dapat dilihat pada Tabel V.

Tabel V. Data uji yang tidak sesuai

No.	Aksara	Target	Hasil Aplikasi
1	ꦗꦸꦩ	jaum	jahum
2	ꦗꦭꦸꦠ	jluat	jluwat
3	ꦗꦲꦁꦢꦶꦁ	jogading	jogadhing
4	ꦒꦸꦫꦲꦫꦶ	ngurah rai	ngurah rahi
5	ꦭꦺꦒꦶꦁ	legian	legyan
6	ꦱꦶꦫꦶꦱꦶ	iriati	hi ri hati
7	ꦠꦲꦶꦤꦺ	taine	tah in nne
8	ꦱꦲꦧꦪꦂꦤ	abian	hab y an

Setelah menyelidiki terjadinya ketidaksesuaian hasil transliterasi, diketahui beberapa penyebab yang menjadi penyebab kesalahan sebagai berikut:

- Perubahan pembacaan pada aksara Bali saat menjadi latin, pada beberapa data uji terdapat teks yang memiliki dua huruf vokal bersamaan pada bentuk latinnya ("au", "ua", "ai", "ia", dll.) sedangkan pada bentuk aksara Balinya dua huruf vokal tersebut tidak beruntun sebagai contoh kata latin dari "jaum" huruf "a" dan "u" beruntun namun dalam aksara Bali kata tersebut akan berupa aksara dengan kata "jahum", sebenarnya aksara Bali mampu membaca dua vokal beruntun seperti ini tanpa merubah pembacaan aksaranya namun kebanyakan penerapan aksara Bali membiarkan perubahan bacaan ini.
- Kata yang tidak terdata pada kamus kata, pada sistem ini terdapat filter 4 dimana pada filter ini teks transliterasi akan diberikan spasi berdasarkan kamus kata yang tersedia karena aslinya aksara Bali tidak memiliki spasi namun aksara latin menggunakannya. Jadi pada beberapa data uji terdapat kata yang tidak

terdata pada kamus kata sehingga pemberian spasi akan tidak sesuai.

4.2.2 Pengujian *Blackbox*

Untuk menguji fungsionalitas dari aplikasi Transliterasi Aksara Bali, dilakukan pengujian *blackbox*. Pada pengujian ini, fungsionalitas aplikasi akan diuji tanpa mengetahui/melihat strukturnya. Pengujian ini dilakukan oleh 10 responden yang merupakan mahasiswa Universitas Mataram yang memiliki perangkat yang berbeda-beda, mulai dari Android dengan versi 6.0 hingga seterusnya. Hasil dari pengujian ini dapat dilihat pada Tabel VI.

Tabel VI. Hasil pengujian *blackbox*.

Halaman transliterasi			
No.	Skenario	Hasil Penguji	
		Berhasil	Tidak
1	Tombol panduan(tombol bergambar tanda tanya di pojok kanan atas layar) berfungsi	10 responden	-
2	Box dropdown aksara dapat menampilkan daftar karakter aksara Bali	10 responden	-
3	Tombol karakter aksara di dalam box dropdown dapat ditekan dan muncul di <i>textbox</i>	10 responden	-
4	Teks aksara Bali yang telah diketik berhasil ditransliterasi/diubah kebentuk huruf latin pada <i>textbox</i> hasil	10 responden	-
5	Tombol tempel(paste) berfungsi	10 responden	-
6	Tombol hapus(delete) berfungsi	10 responden	-
7	Tombol salin(copy) berfungsi	10 responden	-
8	Tombol Auto Spasi berfungsi dan bisa dinyala/dimati kan	10 responden	-
9	Tampilan contoh dapat diakses melalui menu <i>action bar</i>	10 responden	-
10	Tampilan Tentang dapat diakses melalui menu <i>action bar</i>	10 responden	-
Halaman contoh			
1	Teks pada tampilan contoh dapat disalin dan ditempel pada <i>textbox</i>	10 responden	-
Halaman panduan			
1	Tampilan panduan dapat membantu memahami cara mengoperasikan aplikasi	10 responden	-

Dari 10 responden yang mengikuti pengujian *blackbox*, tidak terdapat responden yang menemukan eror atau ketidaksesuaian dari fungsi aplikasi.

5. Kesimpulan dan Saran

Berdasarkan pengujian akurasi yang telah dilakukan, aplikasi Transliterasi Aksara Bali mendapatkan akurasi senilai 99.4% yang terdiri dari 1104 datauji yang bersumber dari kamus Bahasa Bali, nama jalan Bali, dan beberapa artikel aksara Bali. Sedangkan untuk pengujian fungsionalitas yang dilakukan dengan metode *blackbox*, 10 responden dengan perangkat Android versi 6.0 keatas berhasil mengoperasikan aplikasi tanpa terjadi eror sama sekali. Sehingga dapat ditarik kesimpulan bahwa implementasi metode rule based untuk transliterasi Aksara Bali menjadi huruf latin pada Android telah berhasil diimplementasikan dalam rancang bangun aplikasi. Kemudian untuk saran penelitian selanjutnya, dikarenakan penelitian ini yang masih melakukan transliterasi berbasis teks diharapkan kedepannya transliterasi dapat dilakukan melalui gambar atau foto. Selain itu, diharapkan juga untuk menambahkan datauji dan kamus data yang diberikan kepada sistem.

DAFTAR PUSTAKA

- [1] G. Indrawan, N. N. H. Puspita, I. K. Paramarta, and Sariyasa, "LBtrans-bot: A Latin-to-Balinese script transliteration robotic system based on noto sans Balinese font," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 3, pp. 1247–1256, 2018.
- [2] A. Muhajir, "Menjaga Lontar Bali Agar Tak Terlupakan," 2016. [Online]. Available: <https://www.benarnews.org/indonesian/berita/melestarikan-lontar-bali-09302016163916.html>.
- [3] 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*, no. October 2019, pp. 131–136, 2018.
- [4] 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.
- [5] I. B. K. Widiartha and F. Soesianto, "Pembuatan perangkat lunak transliterator Aksara latin ke Aksara Bali dan Aksara Bali ke Aksara latin," *Gajah Mada Univ.*, 2001.

- [6] A. Aranta, F. Bimantoro, and I. P. T. Putrawan, "Penerapan Algoritma Rule Base dengan Pendekatan Hexadesimal pada Transliterasi Aksara Bima Menjadi Huruf Latin," *J. Teknol. Informasi, Komputer, dan Apl. (JTika)*, vol. 2,
- [7] H. GP, Hoetomo U, Miftachul, "Transliterasi Huruf Latin Bahasa Indonesia ke Aksara Jawa Menggunakan Metode Snakecut," *Fortei7*, vol. 2, no. 1, pp. 96–101, 2019.
- [8] B. W. Yohanes, T. Robert, and S. Nugroho, "Sistem Penerjemah Bahasa Jawa-Aksara Jawa Berbasis Finite State Automata," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 6, no. 2, 2017.
- [9] G. Indrawan and I. K. Pramarta, "The Development of Learning Mobile Application of Latin-to-Balinese Script Transliteration," *J. Pekommas*, vol. 4, no. 2, p. 123, 2019.
- [10] 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.
- [11] I. N. Tinggen, *Celah – Celah Kunci Pasang Aksara Bali*. Singaraja: Indra Jaya, 1993.
- [12] I Dewa Ayu Made Sartini, "Pengembangan Text to Digital Image Converter Untuk Dokumen Aksara Bali," *KARMAPATI*, vol. 7, no. June, pp. 1–25, 2012.
- [13] Murtiwiyati and G. Lauren, "Rancang Bangun Aplikasi Pembelajaran Budaya Indonesia Untuk Anak Sekolah Dasar berbasis Android," *J. Ilm.*, vol. 12, p. 2,3, 2013.
- [14] A. Salam and S. Baco, "Rule Base System Untuk Mendukung Surveillance Epidemi," *UIM Makassar*, 2018.
- [15] D. P. Nasional, D. Pendidikan, M. Kejuruan, P. Pengembangan, S. Dan, and P. Smk, "Dasar digital," 2001.
- [16] *Kamus Bali – Indonesia Beraksara Latin dan Bali*. Denpasar: Kerjasama Dinas Kebudayaan Kota Denpasar dengan Badan Pembina Bahasa, Aksara, dan Sastra Bali Provinsi Bali, 2008.
- [17] I. B. A. Sudewa, "Babad Bali - The Balinese Alphabet," *babadbali*, 14-May-2003. [Online]. Available: <http://www.babadbali.com/aksarabali/alphabet-c.htm>. [Accessed: 05-Nov-2020].