

Pengembangan Aplikasi Tranliterasi Aksara Ulu Varian OKU Timur Berbasis Mobile

Andre Fernando¹, Yesi Novaria Kunang², Ilman Zuhri Yadi³, Andri⁴, Susan Dian Purnamasari⁵, Ahmad Rapanie Igama⁶

^{1,2,3,4,5}Sistem Informasi, Universitas Bina Darma, Jl. Jendral Ahmad Yani No. 03, Palembang, 30264, Indonesia

⁶Sastra Indonesia, Universitas Gadjah Mada, Jl. Nusantara No. 01, Yogyakarta, 55281, Indonesia

Info Artikel

Riwayat Artikel:

Received xx - xx - 20xx

Revised xx - xx - 20xx

Accepted xx - xx - 20xx

Abstract – The Ulu script is used by the OKU Timur community, which is one of the ethnic groups in Indonesia located in South Sumatra. The use of the Ulu script has nearly vanished and is being abandoned by the OKU Timur people as times change. This research aims to develop a mobile-based transliteration application for the Ulu script to help preserve it among the OKU Timur community. The application is built using the Flutter framework and the Dart programming language. The development method employed in this project is extreme programming. The final outcome expected is the creation of an easy-to-use application for transliterating Latin text to Ulu script for the wider community.

Keywords: Transliteration, Mobile, Extreme Programming

Corresponding Author:

Yesi Novaria Kunang

Email:

yesinovariakunang@binadarma.ac.id



This is an open access article under the [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/) license.

Abstrak – Penulisan aksara Ulu digunakan oleh masyarakat OKU Timur yang merupakan salah satu suku atau kelompok etnis Indonesia di Sumatera Selatan. Penggunaan aksara Ulu hampir tidak lagi digunakan dan mulai ditinggalkan oleh masyarakat OKU Timur seiring berkembangnya zaman. Penelitian ini dilakukan untuk mengembangkan sebuah aplikasi tranliterasi aksara Ulu berbasis mobile agar dapat membantu dalam melestarikan aksara Ulu OKU Timur. Aplikasi ini dibangun dengan framework Flutter yang menggunakan bahasa pemrograman Dart. Metode yang digunakan dalam pengembangan aplikasi ini adalah extreme programming. Hasil akhir yang diharapkan dengan terciptanya aplikasi yang mudah digunakan untuk melakukan transliterasi teks latin ke aksara Ulu oleh masyarakat luas.

Kata Kunci: Transliterasi, Mobile, Extreme Programming

I. PENDAHULUAN

Indonesia memiliki keanekaragaman suku, budaya dan bahasa dari berbagai daerah yang tersebar di Nusantara. Salah satu dari keanekaragaman tersebut adalah aksara Ulu. Aksara Ulu merupakan bahasa yang digunakan oleh masyarakat OKU Timur, yang mencerminkan nilai-nilai budaya dan tradisi mereka. Namun seiring berjalannya waktu, penggunaan aksara Ulu sudah jarang dipakai oleh masyarakat OKU Timur.

Aksara Ulu merupakan keanekaragaman budaya penting yang dimiliki Indonesia, dikarenakan aksara tersebut banyak terdapat pada naskah kuno, maka dari itu perlu dijaga dan melestarikannya. Salah satu cara untuk melestarikan aksara Ulu ialah dengan menggunakan teknologi, mengingat pada zaman sekarang ini hampir semua aspek kehidupan manusia berhubungan dengan teknologi [1].

Di era kemajuan teknologi digital saat ini, langkah pelestarian Aksara Ulu dapat diwujudkan melalui digitalisasi, di mana aksara tersebut diterapkan dalam perangkat teknologi digital. Maka dari itu untuk memahami tata cara penulisan yang rumit, transliterasi dibutuhkan untuk mengubah teks latin ke aksara Ulu, dengan mengembangkan aplikasi transliterasi menjadi solusi untuk melestarikan serta mempermudah memahami aksara daerah mereka [2]. Tantangan yang muncul dalam penerapan aksara saat ini berkaitan dengan proses transliterasi dan beberapa masalah yang terkait dengan transliterasi dalam penulisan aksara, dikarenakan perbedaan susunan huruf saat diterapkan pada huruf lain. Oleh karena itu, menulis aksara memerlukan keahlian khusus [3].

Pada penelitian sebelumnya, telah dilakukan pembuatan aplikasi transliterasi untuk aksara Ulu varian Besemah [4], namun hingga saat ini, untuk varian OKU Timur belum ada penelitian yang secara khusus melakukan pengembangan aplikasi transliterasi aksara Ulu varian OKU Timur, yang dapat mengubah teks latin menjadi aksara. Berdasarkan permasalahan yang ada, dengan memanfaatkan teknologi untuk membantu melestarikan aksara dengan mengubah teks latin ke aksara Ulu varian OKU Timur. Diharapkan melalui penelitian ini, aplikasi yang dihasilkan efektif dalam melakukan transliterasi dengan benar dan sesuai dengan norma budaya yang berlaku.

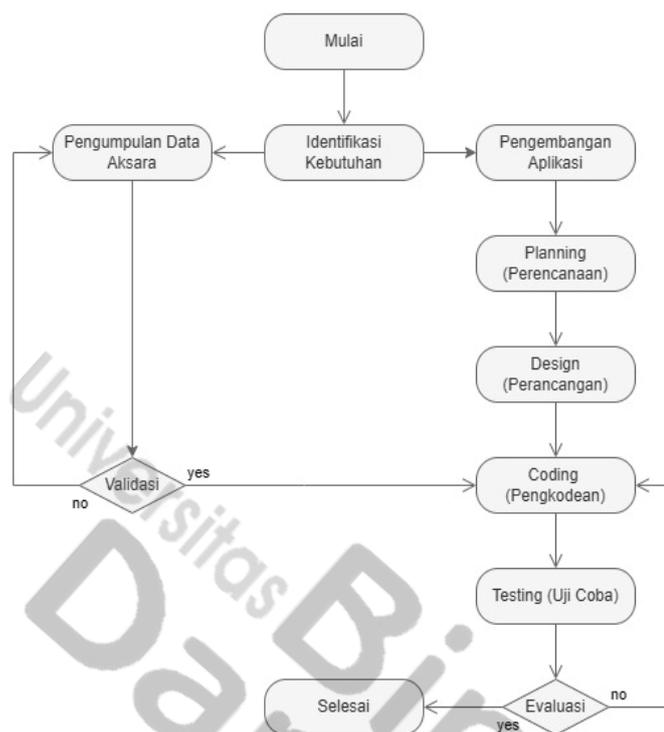
II. METODE

Dalam penelitian ini, flowchart digunakan untuk memvisualisasikan proses pengembangan aplikasi transliterasi. Alur pengembangan dimulai dengan pembuatan API menggunakan Flutter untuk melakukan transliterasi huruf "Aksara OKU Timur (Tanjung Kukuh)" yang akan digunakan dalam aplikasi transliterasi mengacu pada peraturan bupati (perbup) nomor 59 tahun 2023. Di dalam perbup tersebut terdapat 30 aksara, yang terdiri dari 19 aksara utama (konsonan) dan 11 sandangan (vokal). Setelah aksara tersebut dibuat, gambar yang dihasilkan kemudian divalidasi oleh Bapak Ahmad Rapanie, seorang pakar aksara Sumatera Selatan, untuk memastikan keterbacaan dan kesesuaian dengan konteks budaya. Setelah pembuatan huruf selesai, dilakukan implementasi huruf aksara ke dalam aplikasi untuk digunakan dalam proses transliterasi..

π	ζ	ζ	υ	υ	υ
Ka	Ga	Nga	Pa	Ba	Ma
τ	ρ	μ	υ	υ	υ
Ta	Da	Na	Ca	Ja	Nya
ω	π	υ	υ	υ	υ
Ya	Ra	La	Wa	A	Sa
			υ		
			Ha		

No	TANDA BACA	SEBUTAN	BUNYI
1	-√	Hulan	i
2	-√	Arantau	Au
3	-	Kacikoro	O
4	-	Akulu	U
5	-	Incai	Ai
6	-	Bitan	N
7	-	Kadaniah	Il
8	-	Katiluang	Ng
9	-υ	Kakulur	R
10	-√	Nongon	Mati

Setelah itu, dilakukan perencanaan detail mengenai metodologi dan sumber daya yang diperlukan untuk aplikasi. Proses dilanjutkan dengan perancangan sistem berdasarkan rencana yang telah dibuat, diikuti dengan pengkodean untuk menulis kode aplikasi sesuai desain. Setelah aplikasi selesai, uji coba dilakukan untuk memastikan semua fungsi berjalan dengan baik. Hasil penelitian dievaluasi untuk menilai pencapaian tujuan dan mengumpulkan umpan balik.



Dalam proses pengembangan ini dilakukan dengan metode Extreme Programming. Pada metode ini terdapat tahapan yang harus dilakukan sebagai berikut:

A. *Planning*

Pada tahap awal pengembangan, perencanaan dilakukan dengan untuk mengidentifikasi kebutuhan dan tujuan yang harus dipenuhi oleh perangkat lunak yang akan dibuat. Kegiatan ini mengumpulkan informasi serta persyaratan dari pembimbing, serta dilakukan analisis terhadap informasi yang telah dikumpulkan untuk mengidentifikasi masalah dan kebutuhan yang ada.

B. *Design*

Perancangan dilakukan setelah tahap analisis kebutuhan, di mana kegiatan ini berfokus pada merancang struktur dan komponen pada desain antarmuka perangkat lunak. Proses ini bertujuan untuk menciptakan antarmuka yang intuitif dan mudah digunakan. Setiap komponen dirancang secara teliti berdasarkan hasil analisis kebutuhan, sehingga dapat memenuhi harapan pengguna dan mendukung fungsionalitas aplikasi secara keseluruhan. Dengan demikian, perancangan ini menjadi langkah krusial dalam memastikan bahwa perangkat lunak yang dikembangkan tidak hanya efektif, tetapi juga menarik dan mudah diakses oleh penggunanya..

C. *Coding*

Pengkodean dilakukan dengan mengimplementasikan pemrograman berdasarkan hasil dari perancangan desain antarmuka yang sudah dibuat. Penerapan perancangan menggunakan framework Flutter menggunakan bahasa pemrograman Dart. Pada tahap ini proses iterasi dilakukan untuk membangun dan meningkatkan perangkat lunak selama pengembangan.

D. *Testing*

Pada tahap akhir, pengujian dilakukan terhadap aplikasi yang telah dibuat. Pengujian dilakukan dengan pembimbing untuk memastikan bahwa aplikasi yang telah dibuat berjalan dengan baik dan terhindar dari masalah. Pengujian yang telah dilakukan adalah untuk mendapatkan masukan dari pembimbing langsung tentang aplikasi transliterasi aksara. Pada tahap pengujian, iterasi dilakukan selama pengembangan untuk merespon perubahan dengan cepat menyempurnakan dan meningkatkan perangkat lunak.

III. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini menggunakan penerapan metode Extreme Programming (XP) dalam pengembangan aplikasi transliterasi melalui tahapan Planning (Perencanaan), Design (Perancangan), Coding (Pengkodean), dan Testing (Uji Coba).

A. Planning (Perencanaan)

Perencanaan pada pengembangan aplikasi dilakukan identifikasi masalah bahwa penggunaan Aksara Ulu di masyarakat OKU Timur semakin menurun, dan belum adanya aplikasi transliterasi untuk menerjemahkan aksara tersebut. Selain itu, proses transliterasi Aksara Ulu menghadapi tantangan signifikan, terutama terkait dengan perbedaan susunan huruf dan kompleksitas penulisan yang memerlukan keahlian khusus. Meskipun sudah ada aplikasi transliterasi untuk Aksara Ulu varian Besemah, belum ada penelitian atau aplikasi yang secara khusus mengembangkan transliterasi untuk aksara Ulu varian OKU Timur. Oleh karena itu, diperlukan aplikasi seperti transliterasi, guna melestarikan Aksara Ulu OKU Timur. Berikut merupakan analisis kebutuhan berdasarkan identifikasi masalah untuk membangun aplikasi transliterasi berbasis mobile:

1. Pengguna dapat melakukan transliterasi dari teks latin ke Aksara Ulu OKU Timur.
2. Web server dapat menyimpan dan mengelola data Aksara Ulu OKU Timur.
3. Desain tampilan aplikasi dibuat sederhana agar dapat digunakan oleh orang awam.
4. Mobile flutter dapat menggunakan API untuk berinteraksi dengan server dan mengambil data atau fitur transliterasi.
5. Web server memungkinkan aplikasi mobile flutter berkomunikasi dengan backend dan mengakses berbagai layanan dan data.
6. Mobile flutter dapat menampilkan hasil dari transliterasi teks latin berupa gambar dan deskripsi aksara.
7. Normalisasi aksara dilakukan agar membantu sistem memahami dan memproses teks dengan akurat.

B. Design (Perancangan)

Perancangan sistem dilakukan pada saat pengembangan tampilan, antarmuka pengguna aplikasi yang dibuat sederhana agar mudah digunakan. Berikut rancangan wireframe tampilan antarmuka pengguna aplikasi transliterasi.

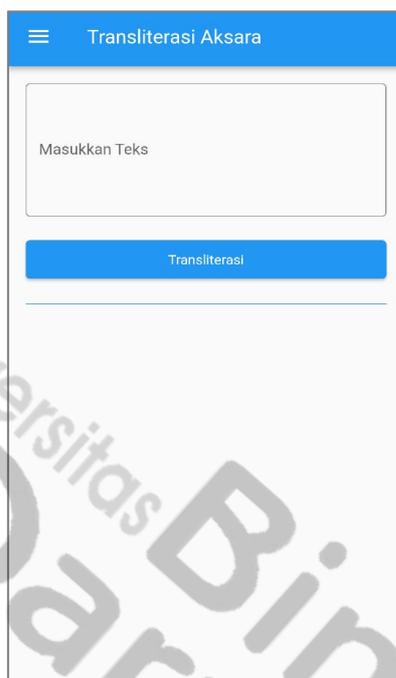


Setelah tahapan-tahapan pengembangan sistem selesai, hasil implementasi dari aplikasi yang telah dikembangkan dapat ditinjau. Berikut hasil tampilan dari aplikasi.

1. Halaman Transliterasi.

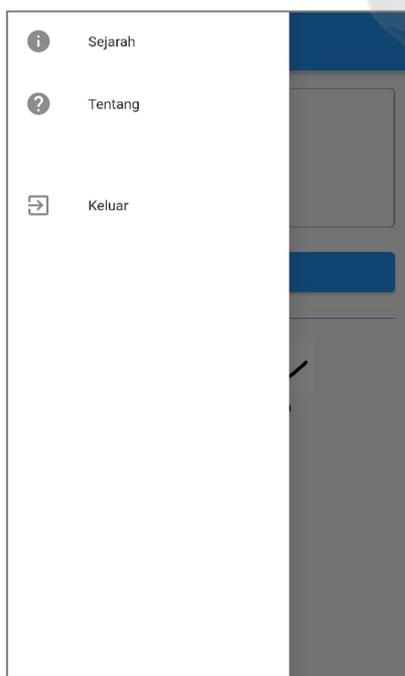
Tampilan aplikasi pada halaman untuk melakukan transliterasi dari Bahasa Indonesia ke aksara Ulu OKU Timur dirancang dengan sederhana dan intuitif, sehingga memudahkan pengguna dalam berinteraksi. Di bagian atas halaman, terdapat kolom teks yang memungkinkan pengguna untuk memasukkan kalimat atau kata-kata yang ingin ditransliterasi. Setelah pengguna mengetikkan teks, mereka dapat menekan tombol "Transliterasi" yang berfungsi untuk mengirimkan permintaan ke server. Hasil transliterasi akan ditampilkan di bawah kolom input

dalam bentuk aksara Ulu, lengkap dengan gambar dan keterangan yang menjelaskan setiap karakter. Dengan antarmuka yang responsif dan mudah digunakan.



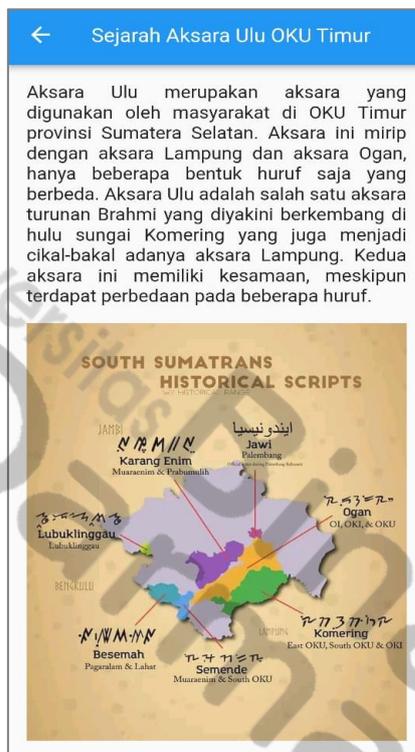
2. Halaman Sidebar.

Tampilan sidebar terdapat beberapa menu untuk memudahkan navigasi pengguna. Di dalam sidebar, terdapat menu "Sejarah Aksara Ulu OKU Timur" yang memberikan informasi mendalam mengenai asal-usul dan perkembangan aksara tersebut, sehingga pengguna dapat lebih memahami konteks budaya di baliknya. Selain itu, terdapat juga menu "Tentang Aplikasi" yang menjelaskan tujuan dan fungsi aplikasi ini, serta cara penggunaannya. Menu terakhir adalah opsi "Exit" yang memungkinkan pengguna untuk keluar dari aplikasi.



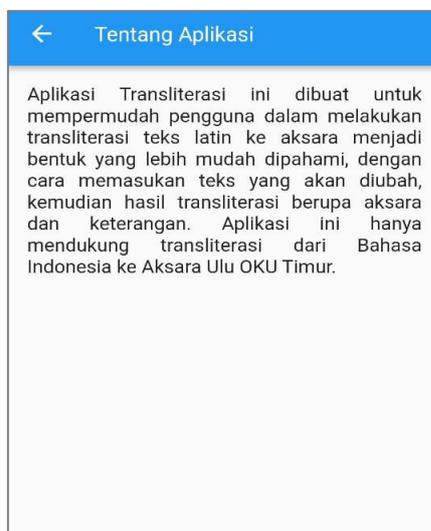
3. Halaman Sejarah.

Pada menu "Sejarah Aksara Ulu OKU Timur," pengguna dapat menemukan informasi yang mendalam mengenai asal-usul dan perkembangan aksara ini. Halaman ini dirancang untuk memberikan wawasan tentang bagaimana aksara Ulu OKU Timur muncul dan bagaimana ia telah berkembang seiring waktu dalam konteks budaya masyarakat setempat.



4. Halaman Tentang Aplikasi.

Pada menu "Tentang Aplikasi," pengguna dapat menemukan penjelasan mengenai tujuan dan fungsi aplikasi transliterasi ini. Aplikasi ini dirancang untuk mempermudah pengguna dalam melakukan transliterasi teks latin ke aksara. Pengguna cukup memasukkan teks yang ingin diubah, dan aplikasi akan menampilkan hasil transliterasi berupa aksara beserta keterangan yang relevan. Penting untuk dicatat bahwa aplikasi ini hanya mendukung transliterasi dari bahasa Indonesia ke Aksara Ulu Oku Timur.



C. Coding (Pengkodean)

Pengkodean dilakukan dengan mengimplementasikan rancangan sistem dan arsitektur yang telah disusun dalam bentuk program aplikasi. Pengembangan antarmuka pengguna aplikasi transliterasi menggunakan bahasa pemrograman Dart dengan framework Flutter. Pada backend, Laravel digunakan sebagai web server, dengan MySQL sebagai sistem manajemen basis data.

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
import 'package:flutter/services.dart';

void main() {
  runApp(MyApp());
}
```

Bagian ini mengimpor paket yang diperlukan untuk aplikasi Flutter ('material.dart') dan paket untuk melakukan permintaan HTTP ('http'). Fungsi 'main()' menjalankan aplikasi. Berikut adalah penjelasan rinci dengan fokus pada bagian fungsi yang di implementasikan sebagai kode:

1. MyApp Widget

MyApp adalah widget stateless yang berfungsi sebagai titik awal aplikasi. Ini mengatur judul aplikasi, tema, dan halaman utama yang ditampilkan ('MyHomePage').

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
```

2. MyHomePage StatefulWidget

MyHomePage adalah sebuah widget stateful yang berfungsi untuk memungkinkan perubahan User Interface (UI) berdasarkan interaksi yang dilakukan oleh pengguna. Dalam widget ini, keadaan (state) dikelola oleh kelas yang bernama '_MyHomePageState'. Dengan menggunakan widget stateful, aplikasi dapat merespons input pengguna dengan memperbarui tampilan secara dinamis sesuai dengan perubahan yang terjadi.

```
class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() =>
  _MyHomePageState();
}
```

3. Controller Input Text

Fungsi '_inputController' adalah sebuah kontroler untuk mengelola dan mengambil teks yang dimasukkan oleh pengguna melalui elemen 'TextField'. Dengan adanya kontroler ini, aplikasi dapat memantau setiap perubahan yang terjadi pada input pengguna secara langsung, sehingga memungkinkan pengambilan data yang lebih tepat untuk diproses atau digunakan dalam fungsi lain di dalam aplikasi.

```
final TextEditingController _inputController =  
TextEditingController();
```

4. Daftar Hasil Transliterasi

Daftar hasil ‘_results’ adalah sebuah list yang menyimpan hasil transliterasi yang diperoleh dari API dalam bentuk peta (map). Setiap item dalam list ini berisi dua elemen penting: URL gambar (‘file’) dan keterangan (‘label’). Dengan menggunakan daftar ini, aplikasi dapat menampilkan hasil transliterasi secara terstruktur, memudahkan pengguna untuk mengakses informasi yang relevan.

```
List<Map<String, dynamic>> _results = [];
```

5. Fungsi ‘_transliterate()’ untuk Memproses Transliterasi.

Fungsi _transliterate() bertugas untuk melakukan permintaan POST ke API transliterasi. Fungsi ini mengirimkan teks yang dimasukkan oleh pengguna ke API dan menerima respons yang berisi hasil transliterasi. Untuk melaksanakan ini, Uri.parse(...) digunakan untuk mengubah URL API menjadi URI yang dapat digunakan dalam permintaan HTTP. Selain itu, bagian headers mencakup informasi Content-Type dan apikey sebagai 25 bagian dari permintaan. Pada bagian body, teks input yang dimasukkan diencode dalam format JSON untuk dikirimkan.

```
Future<void> _transliterate() async {  
  final response = await http.post(  
    Uri.parse('https://3b0a-158-140-165-60.ngrok  
    free.app/api/translate'),  
    headers: {  
      'Content-Type': 'application/json',  
      'apikey': '1e72d71c46bca27391651d51ada34145'  
    },  
    body:  
    jsonEncode({'text':_inputController.text}),  
  );
```

6. Penanganan Respon dari API.

Dalam proses penanganan respons dari API, terdapat beberapa langkah penting yang harus dilakukan untuk memastikan data yang diterima dapat diproses dengan benar. Pertama, pernyataan if (response.statusCode == 200) digunakan untuk memeriksa apakah permintaan yang diajukan berhasil. Jika berhasil, fungsi jsonDecode(response.body) akan mengonversi respons yang berbentuk 26 JSON menjadi objek Dart yang dapat digunakan. Dari objek ini, aksaraResult menjadi daftar hasil transliterasi yang diterima dari respons API. Selanjutnya, dengan memanggil setState(), nilai _results diperbarui untuk menyimpan hasil transliterasi tersebut, yang secara otomatis menyebabkan antarmuka pengguna diperbarui agar menampilkan hasil terbaru.

```
if (response.statusCode == 200) {  
    final data = jsonDecode(response.body);  
    final List<dynamic> aksaraResult =  
data['data']['aksara_result'];  
  
    setState(() {  
        _results = aksaraResult.map((item) {  
            return {  
                'file': item['file'],  
                'label': item['keterangan'],  
            };  
        }).toList();  
    });  
}
```

7. Membangun UI di 'build()'

Dalam membangun User Interface (UI) di dalam metode build(), terdapat beberapa komponen penting yang berfungsi untuk memfasilitasi interaksi pengguna. Pertama, terdapat TextField yang berfungsi sebagai tempat bagi pengguna untuk memasukkan teks yang akan ditransliterasi. Setelah itu, pengguna dapat menekan ElevatedButton, yang akan memanggil fungsi _transliterate() untuk memulai proses transliterasi. Jika ada hasil yang diperoleh, hasil tersebut akan ditampilkan dalam SingleChildScrollView yang memungkinkan pengguna untuk menggulir secara horizontal. Untuk menampilkan gambar dari hasil transliterasi, digunakan komponen Image.network, yang mengambil gambar dari URL yang diberikan. Terakhir, informasi tambahan mengenai hasil transliterasi ditampilkan dengan menggunakan Text(result['label']), yang menyajikan keterangan atau label yang sesuai.

```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: Text('Transliterasi Aksara'),  
    ),  
    drawer: Drawer(  
      width: 250,  
      child: Container(  
        color: Colors.white,  
        child: ListView(  
          padding: EdgeInsets.zero,  
          children: <Widget>[  
            SizedBox(height: 20),  
            ListTile(  
              leading: Icon(Icons.info),  
              title: Text('Sejarah'),  
              onTap: () {  
                Navigator.pop(context); // Tutup drawer  
                _navigateTo('Sejarah'); // Navigasi ke  
halaman Sejarah  
              },  
            ),  
            ListTile(  
              leading: Icon(Icons.help),  
              title: Text('Tentang'),  
              onTap: () {
```

8. Proses Transliterasi.

Proses transliterasi merupakan langkah penting dalam aplikasi ini, yang menggabungkan interaksi pengguna dengan backend API untuk menghasilkan dan menampilkan hasil secara real-time di aplikasi Flutter. Di sisi backend, yang menggunakan Laravel sebagai server web, dilakukan proses pemenggalan suku kata. Fungsi ini memecah kata menjadi suku kata, serta memisahkan konsonan dan vokal sesuai dengan aturan yang berlaku. Proses pemecahan kata ini dilakukan dengan membagi kata menjadi bagian yang lebih kecil berdasarkan urutan konsonan dan vokal. 30 Jika kata yang dimasukkan tidak ditemukan, sistem akan melakukan pemetaan karakter terhadap aksara dasar.

```
global vokal
result = []
last = ""
i = 0
for suku in listsuku:
    if len(suku)==1 and i>0:
        if suku in vokal:
            result += [suku]
        elif listsuku[i-1][-1] in vokal:
            if i<len(listsuku)-1 and
len(listsuku[i+1])==1 and listsuku[i+1][0] in vokal:
                result += [suku]
            else:
                result[-1] = result[-1] + suku
        else:
            result += [suku]
    i += 1
    last = suku
return result
```

Fungsi ini bertugas untuk memecah kata menjadi suku-suku kata dengan memisahkan konsonan dan vokal sesuai dengan aturan yang berlaku. Kata-kata dipecah menjadi bagian yang lebih kecil berdasarkan urutan konsonan dan vokalnya. Jika suatu kata tidak ditemukan, sistem akan melakukan pemetaan karakter ke aksara dasar yang sesuai

```
['huruf' => ['zang',], 'aksara' => 'jang'],
['huruf' => ['zan',], 'aksara' => 'jan'],
];

foreach ($itemOfArray as $keyKalimat => $kalimat) {
    foreach (array_column($kamusFilter, 'huruf') as
$keyFilter => $filter) {
        if(in_array($kalimat, $filter)){
            $itemOfArray[$keyKalimat] =
$kamusFilter[$keyFilter]['aksara'];
        }
    }
}
}
```

Penting untuk dicatat bahwa dalam aksara ini, beberapa huruf Latin seperti F, V, P, Q, dan Z tidak dicakup. Oleh karena itu, huruf-huruf tersebut akan diganti dengan karakter aksara Ulu yang lain; misalnya, F dan V akan diganti dengan karakter P, X dan Q akan diganti dengan karakter K, serta Z akan diganti dengan karakter S. Kode ini berfungsi untuk melakukan penggantian karakter tersebut agar sesuai dengan kaidah aksara Ulu.

D. Testing (Uji Coba)

Pengujian sistem dilakukan untuk mengidentifikasi kesalahan yang mungkin muncul saat aplikasi dijalankan serta memastikan bahwa sistem pada aplikasi yang dikembangkan sesuai dengan kebutuhan dan berjalan dengan optimal. Metode pengujian yang digunakan adalah black box testing, yang bertujuan untuk menguji aspek fungsional dari perangkat lunak yang dirancang.

NO	Fungsi	Pengujian	Hasil
1.	Request	Mengirim permintaan transliterasi dari aplikasi Flutter ke web server menggunakan API.	Berhasil
2.	Response	Web server mengembalikan hasil transliterasi.	Berhasil
3.	Transliterasi	Flutter menampilkan hasil berupa gambar dan deskripsi aksara dari response web server menggunakan API.	Berhasil
4.	Data Aksara	Semua data aksara dapat tampil pada saat Flutter mengirim permintaan ke web server menggunakan API.	Berhasil
5.	Hasil Tranliterasi Aksara	Hasil dari transliterasi teks latin ke aksara sesuai dengan gambar dan deskripsi aksara pada database web server.	Berhasil

Pengujian API dilakukan pada file Python yang telah dibuat, di mana file tersebut berisi data yang akan diterjemahkan dan kemudian dijalankan untuk melihat hasilnya. Setelah proses pengujian, data aksara yang dihasilkan akan ditampilkan ketika aplikasi Flutter mengirimkan permintaan ke web server menggunakan API. Hasil dari transliterasi teks Latin ke aksara akan sesuai dengan gambar dan deskripsi aksara yang tersimpan dalam database di web server.

```
{ "status": "SUCCESS", "timestamp": "2024-08-27T18:25:34.791477Z", "text": "saya", "data":  
  { "split_result": ["sa", "ya"], "aksara_result":  
    [ { "id": 181, "file": "http://127.0.0.1:8000/data_aksara/sa.png", "keterangan": "sa" },  
      { "id": 217, "file": "http://127.0.0.1:8000/data_aksara/ya.png", "keterangan": "ya" } ] } }
```

Pengujian ini bertujuan untuk memastikan bahwa API berfungsi dengan baik dalam menerjemahkan data dari format teks Latin ke aksara yang diinginkan. Dengan mengirimkan permintaan melalui Flutter, aplikasi akan menerima respons dari server yang berisi hasil transliterasi. Respons ini tidak hanya mencakup aksara yang dihasilkan, tetapi juga gambar dan deskripsi yang relevan, sehingga pengguna dapat memahami konteks dari aksara yang ditampilkan.



IV. SIMPULAN

Berdasarkan hasil dari penelitian untuk pengembangan aplikasi transliterasi Aksara Ulu varian OKU Timur. Aplikasi Flutter ini memungkinkan pengguna mentransliterasi teks menjadi berbagai aksara melalui API eksternal. Pengguna cukup memasukkan teks, dan aplikasi akan memprosesnya, menampilkan hasil transliterasi dalam bentuk gambar aksara dan keterangannya. Aplikasi ini menggabungkan antarmuka yang interaktif dengan backend API, memperbarui tampilan UI secara dinamis berdasarkan respons API. Keunggulannya terletak pada kemudahan penggunaan dan responsivitasnya, aplikasi diakses dan digunakan menggunakan perangkat mobile smartphone, aplikasi dapat berjalan lancar khususnya dalam melakukan transliterasi. Hal ini telah diuji menggunakan black box testing dan berhasil melakukan tahap uji coba secara fungsional, namun keamanan API key, penanganan kesalahan, dan optimasi performa masih bisa ditingkatkan. Secara keseluruhan, aplikasi ini memberikan dasar yang kuat untuk pengembangan fitur transliterasi yang lebih lanjut.

DAFTAR PUSTAKA

- [1] Abdurrahim, R., Ayyubi, A., Khovivil Anam, M., & Permatasari, H. (2023). Pengujian Sistem Informasi Penjualan Menggunakan Pengujian Black Box Testing dan White Box Testing. *SENATIB*.
- [2] Andesa, K., Fachzevi, A., Nasution, T., & Herwin. (2023). Implementasi Metode AES pada Aplikasi Chat Menggunakan Flutter. *SATIN - Sains Dan Teknologi Informasi*, 9(2), 216–226. <https://doi.org/10.33372/stn.v9i2.10751>. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in *Magnetism*, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- [3] Aranta, A., Bimantoro, F., & Putrawan, I. P. T. (2020). Penerapan Algoritma Rule Base dengan Pendekatan Hexadesimal pada Transliterasi Aksara Bima Menjadi Huruf Latin. *Jurnal Teknologi Informasi, Komputer, Dan Aplikasinya (JTika)*, 2(1), 130–141. <https://doi.org/10.29303/jtika.v2i1.96R>. Nicole, “Title of paper with only first word capitalized,” *J. Name Stand. Abbrev.*, in press.
- [4] Faust, S. (2020). Using Google’s Flutter Framework for the Development of a Large-Scale Reference Application. <https://nbn-resolving.org/urn:nbn:de:hbz:832-epub4-14989>.
- [5] Mahmud, M., & Kunang, Y. N. (2020). Pengembangan Aplikasi Pengenalan Aksara Komerling Menggunakan Metode Deep Learning Berbasis Android. *Journal of Computer and Information Systems Ampera*, 1(2), 99–111. <https://doi.org/10.51519/journalcisa.v1i2.38>
- [6] Nurkholis, A., Susanto, E. R., & Wijaya, S. (2021). Penerapan Extreme Programming dalam Pengembangan Sistem Informasi Manajemen Pelayanan Publik. *J SAKTI (Jurnal Sains Komputer Dan Informatika)*, 5(1), 124–134.
- [7] Permata Sari, T., & Novaria Kunang, Y. (2023). Pengembangan Aplikasi Transliterasi Teks Latin ke Aksara Ulu (Komerling) Berbasis Web. *Jurnal PROCESSOR*, 18(2). <https://doi.org/10.33998/processor.2023.18.2.852>.
- [8] Praheto, B. E., Andayani, M. R., & Wardani, N. E. (2020). The effectiveness of interactive multimedia in learning Indonesian language skills in higher education. *Rupkatha J. Interdiscip. Stud. Humanity*, 12(1), 1–11.
- [9] Pratama, P. W., Aranta, A., & Bimantoro, F. (2021). Rancang Bangun Aplikasi Transliterasi Aksara Latin menjadi Aksara Sasak Menggunakan Algoritma Rule Based Berbasis Android. *Jurnal Teknologi Informasi, Komputer, Dan Aplikasinya (JTika)*, 3(2), 232–243.