

# Transliterasi Aksara Bali menjadi Huruf Latin pada Citra Tulisan Tangan

I Made Arya Dwisada<sup>a1</sup>, I Gusti Agung Gede Arya Kadyanan<sup>a2</sup>, I Dewa Made Bayu Atmaja Darmawan<sup>a3</sup>, Cokorda Rai Adi Prammartha<sup>a4</sup>, I Komang Ari Mogi<sup>a5</sup>, I Gusti Ngurah Anom Cahyadi Putra<sup>a6</sup>

<sup>a</sup>Program Studi Informatika, Universitas Udayana  
Jl. Kampus Bukit Jimbaran, Gedung BF Jimbaran, Badung, Bali 80361, Indonesia  
<sup>1</sup>aryadwisada46@gmail.com  
<sup>2</sup>gungde@unud.ac.id  
<sup>3</sup>dewabayu@unud.ac.id  
<sup>4</sup>cokorda@unud.ac.id  
<sup>5</sup>arimogi@unud.ac.id  
<sup>6</sup>anom.cp@unud.ac.id

## Abstract

*The application of image recognition has been widely used in various fields in daily life, one of which is the introduction of useful and useful text or characters in increasing work productivity. Character recognition can be used in the process of transliterating characters, such as translating Balinese script into Latin letters. To perform the previous character transfer, image processing was needed such as preprocessing line segmentation data and character segmentation, using the Image Centroid Zone + Zone Centroid Zone and Moment Invariant feature extraction method and using the Support Vector Machine method. The introduction of Balinese script in handwriting using the Moment Invariant and Zoning feature extraction methods (Image Centroid Zone and Zone Centroid Zone) and the Support Vector Machine classification method resulted in the recognition accuracy in model formation of 90.3%, and the average accuracy generated after testing the system using test data is 82.936%. The making of the rule base for Balinese script is formed using a series of conditions using if-else that adjusts to the rules of reading Balinese script. The conversion of Balinese script in handwriting into Latin letters in this study resulted in an average accuracy of 82.182% of the 6 image data used in system testing.*

**Keywords:** *Transliteration, Segmentation, Recognition, Moment Invariant, Zoning, Support Vector Machine*

## 1. Pendahuluan

Aksara Bali merupakan salah satu aksara tradisional yang ada di Indonesia. Aksara Bali adalah aksara yang berkembang di Pulau Bali yang biasanya digunakan untuk menuliskan bahasa Bali, Sansekerta dan Kawi, bahkan pada beberapa dokumen yang ada aksara Bali digunakan untuk menulis bahasa Sasak dan Melayu. Aksara Bali adalah aksara hasil perkembangan aksara dan budaya yang disebarkan oleh orang-orang India yang juga melakukan penyebaran agama dan kebudayaan. Arah penulisan aksara Bali ditulis dari kiri ke kanan dan ditulis tanpa spasi antarkata. Aksara Bali juga memiliki beberapa atribut dan tanda baca yang digunakan untuk menyusun suatu kata atau kalimat.

Aksara Bali secara umum dibagi menjadi dua jenis yaitu aksara biasa, dan aksara suci. Aksara yang biasa digunakan yaitu aksara swalalita yang terdiri dari aksara suara, Aksara wresastra dan aksara wayah. Aksara Bali biasanya digunakan untuk menulis dokumen nyanyian, cerita, resep, perjanjian, cacatan, naskah, dan penanda tempat atau bangunan dan lainnya. Penggunaan aksara Bali salah satunya yaitu digunakan untuk menuliskan kakawin yang mengandung bahasa Kawi maupun bahasa Sansekerta. Penulisan aksara Bali pada kakawin menggunakan aksara swalalita. Alih aksara Bali penting dilakukan untuk mempermudah mempelajari dokumen yang mengandung aksara Bali tersebut.

Alih aksara juga dapat berguna untuk edukasi. Saat ini sudah ada beberapa sistem dan aplikasi pembelajaran aksara Bali, dan ada juga sistem atau aplikasi alih aksara latin menjadi aksara Bali, namun belum banyak sistem yang mengalihaksarakan aksara Bali menjadi huruf latin sehingga perlu dilakukan pengembangannya.

Perkembangan teknologi pada saat ini menghasilkan teknologi pengenalan pola yang dapat diterapkan untuk melakukan alih aksara Bali. Proses pengenalan pola terhadap aksara maka dapat dihasilkan karakter yang dikenali. Hasil dari karakter tersebut kemudian dilakukan pelabelan dan menghasilkan kumpulan karakter yang telah dikenali. Kemudian dari susunan aksara yang telah melalui

proses pelabelan maka dapat dilakukan alih aksara dengan aturan atau rule penulisan dan bacaan aksara itu. Beberapa proses tersebut dapat diterapkan untuk melakukan alih aksara Bali menjadi huruf latin.

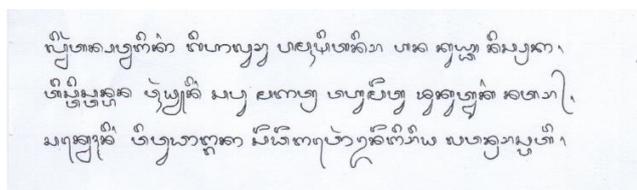
Citra karakter diperoleh dengan melakukan segmentasi baris terlebih dahulu. Tujuan dari segmentasi baris adalah membagi citra menjadi beberapa potongan baris, dan ketika sudah mendapatkan hasil segmentasi baris maka data citra tersebut dapat digunakan untuk menentukan potongan citra karakter aksara. Dilakukan segmentasi baris ini akan membuat proses segmentasi karakter aksara lebih optimal. Terdapat penelitian mengenai segmentasi baris yang menggunakan metode *A\* Path Planning* [1]. Penelitian ini menghasilkan hasil yang akurat dalam melakukan proses segmentasi baris pada dataset manuskrip dan bekerja sangat efektif jika komponen dua garis tidak tumpang tindih. Penelitian sebelumnya juga melakukan segmentasi karakter aksara Bali dengan menerapkan metode proyeksi horizontal untuk menghasilkan baris yang tersegmentasi dan proyeksi vertikal yang menghasilkan batas karakter, sehingga menghasilkan segmentasi karakter aksara Bali[2]. Penelitian yang melakukan pengenalan pola tulisan tangan suku aksara Sasak menggunakan metode ekstraksi fitur *moment invariant* dan metode *Support Vector Machine* untuk proses klasifikasi dan menghasilkan akurasi yang baik yang dipengaruhi oleh jumlah fitur, semakin banyak fitur yang digunakan maka akan semakin baik akurasi klasifikasi [3].

Berdasarkan pemaparan diatas, maka pada penelitian kali ini akan dilakukan penelitian alih aksara Bali pada citra tulisan tangan menjadi huruf latin yang datanya mengandung aksara swalalita dengan melakukan segmentasi karakter, melakukan pengenalan dan *labeling* kemudian melakukan penyusunan huruf latin sesuai dengan aturan penulisan aksara Bali dengan mengambil urutan data karakter yang dikenali yang dimasukkan ke dalam proses yang dibentuk dengan berbasis aturan atau rule base.

## 2. Metode Penelitian

### 2.1. Data Penelitian

Data penelitian kali ini berupa tulisan tangan aksara Bali yang merupakan tulisan ulang dari buku kakawin Arjuna Wiwaha oleh Dinas Pendidikan Dasar Provinsi Daerah Tingkat Satu Bali 1988, pupuh ke tiga Raga Kusuma, bait ke enam bari pertama sampai baris ketiga. Jenis aksara yang dituliskan adalah aksara swalalita. Data yang diambil sebanyak 66 data, pengambilan data dilakukan dengan memberikan kertas HVS berukuran A4 dan alat tulis berupa pena kepada 22 orang responden yang memiliki kemampuan menuliskan aksara Bali yang baik, kertas yang diberikan kepada responden adalah kertas telah di berikan garis bantu yang dibuat dengan goresan pensil untuk mensejajarkan baris tulisan pada wilayah yang telah ditentukan, penggunaan pensil untuk pembuatan garis adalah agar garis bantu dapat dihilangkan ketika data akan di *scan*. Data citra yang digunakan merupakan gambar dengan format .png. Setelah di *scan* dilakukan proses pemotongan secara manual sebelum digunakan.

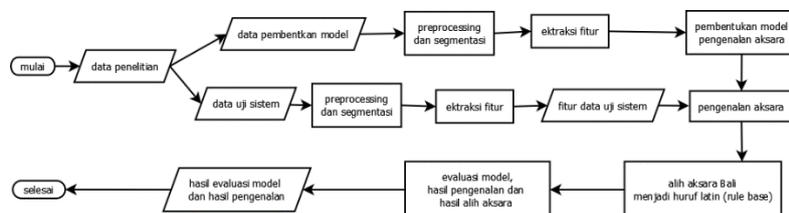


Gambar 1. Contoh data citra tulisan tangan

Gambar 1 merupakan salah satu data penelitian yang digunakan dalam penelitian kali ini

## 2.2. Desain Penelitian

Rancangan proses yang akan diimplementasikan dalam penelitian mengenai alih aksara Bali pada citra tulisan tangan menjadi huruf latin dengan segmentasi karakter aksara secara umum akan dijabarkan seperti diagram alir (*flowchart*) pada Gambar 2.



Gambar 2. Alur penelitian

Data penelitian yang totalnya berjumlah 66 citra akan dibagi menjadi dua yaitu data yang digunakan dalam pembentukan model klasifikasi sebanyak 60 citra dan data untuk pengujian sistem sebanyak 6 citra. Data yang digunakan pembentukan model terlebih dahulu masuk dalam proses *preprocessing* dan segmentasi untuk menghasilkan citra karakter, lalu citra karakter tersebut akan masuk ke dalam proses ekstraksi fitur yang bertujuan untuk menghasilkan fitur yang akan digunakan untuk pembentukan model pengenalan. Sedangkan untuk data uji akan masuk juga ke dalam proses *preprocessing* dan segmentasi untuk mendapatkan citra karakter, kemudian citra karakter tersebut akan masuk ke dalam proses ekstraksi fitur dan fitur yang dihasilkan akan masuk ke dalam proses pengenalan serta hasil pengenalan akan diproses untuk menghasilkan huruf latin yang merupakan hasil alih aksara.

## 2.3. Metode A\* Path Planning

Algoritma *A\* Path Planning* menggunakan fungsi heuristik untuk menemukan dan solusi yang tepat untuk mencapai keadaan tujuan dengan menggabungkan biaya jalur aktual dan fungsi heuristik untuk memperkirakan biaya dari *state* awal ( $s_1$ ) ke *state* tujuan ( $s_n$ ) [1]. Jika  $s_1^a, s_2^a, \dots, s_n^a$ , adalah urutan *state* yang dilintasi jalur  $p^a$  maka total biaya perjalanan terendah dihitung dengan persamaan berikut

$$. p^* = \arg \min_{p^a} \sum_{i=1}^{n-1} C(s_i^a, s_{i+1}^a) \quad (1)$$

Keterangan:  $C(s_i, s_j)$ : biaya perjalanan dari *state*  $s_i$  sampai ke *state*  $s_j$

Untuk melakukan perhitungan biaya perjalanan dari *state* satu ke *state* yang lainnya algoritma *A\* Path Planning* menggunakan kombinasi fungsi-fungsi biaya sebagai berikut:

- Fungsi *Ink Distance Cost* untuk mengontrol jalur agar tetap berada diantara piksel *foreground* yang dipisahkan yaitu fungsi  $D(n)$  dan  $D(n)^2$ .

Fungsi biaya jarak *foreground*  $D(n)$  yang melewati jalur *state*  $n$  adalah sebagai berikut:

$$D(n) = \frac{1}{1 + \min[d(n, n_{yu}), d(n, n_{yd})]} \quad (2)$$

Fungsi  $D(n)^2$  digunakan untuk menghasilkan biaya yang jauh lebih tinggi untuk mendekati nilai piksel dibandingkan dengan menjauh dari piksel hitam. Fungsi ini adalah sebagai berikut:

$$D(n)^2 = \frac{1}{1 + \min[d(n, n_{yu}), d(n, n_{yd})]^2} \quad (3)$$

Keterangan:

$d(n, n_{yu})$  dan  $d(n, n_{yd})$ : jarak terdekat antara *state*  $n$  dengan *foreground* di atasnya dan *foreground* dibawahnya.

- b. Fungsi Map-Obstacle Cost  $M(n)$  adalah fungsi yang digunakan agar *path* tidak melewati piksel *foreground*. Jika *state* berada pada piksel hitam maka fungsi akan me-*return* 1 dan jika tidak maka akan me-*return* 0.
- c. Fungsi *Vertical Cost*  $V(n)$  digunakan untuk mencegah jalur menyimpang dari posisi  $y$  dari *state* awal dan *state* selanjutnya, berikut adalah fungsi *Vertical Cost*  $V(n)$ .  
$$V(n) = \text{abs}(n_y - n_y^{\text{start}}) \quad (4)$$

Keterangan:

$n_y$  : posisi  $y$  dari *state* saat ini

$n_y^{\text{start}}$  : posisi  $y$  dari *state* awal

- d. Fungsi *Neighbor Cost Function*  $N(s_i, s_j)$  adalah fungsi yang digunakan untuk menghitung jalur terpendek antara *state* awal dengan *state* tujuan. Nilai dari  $N(s_i, s_j)$  akan menjadi 14 ketika berpindah ke *state* baru untuk arah diagonal, dan akan bernilai 10 untuk vertikal dan horizontal.

Gabungan dari fungsi biaya menghasilkan rumus algoritma *A\* Path Planning* adalah sebagai berikut:  
$$C(s_i, s_j) = c_d D(s_i) + c_{d2} D(s_i)^2 + c_m M(s_i) + c_v V(s_i) + c_n N(s_i, s_j) \quad (5)$$

#### 2.4. Metode *Connected Component Labeling*

Algoritma *Connected Component Labeling* secara umum digunakan dalam proses yang berhubungan dengan deteksi objek [4]. Pemberian label berbeda setiap karakter maka karakter yang satu dengan yang lain dapat dipisahkan berdasarkan label yang dimilikinya. Hasil pelabelan ditentukan pula oleh jumlah konektivitas yang digunakan pada citra 2 dimensi yaitu 4-Konektivitas dan 8-Konektivitas. Metode ini banyak digunakan untuk melakukan proses segmentasi citra seperti pada penelitian [5][6].

#### 2.5. Metode *Zoning*

*Zoning* merupakan salah satu metode ekstraksi ciri yaitu dengan cara membagi citra menjadi beberapa zona yang sama untuk dikenali ciri dari setiap karakter aksara. Keunggulan dari *zoning* yaitu merupakan metode pencirian yang sederhana, kompleksitas yang rendah dan memiliki perhitungan yang cepat dalam mengekstraksi ciri suatu karakter [7], serta algoritma ini mampu meningkatkan akurasi dalam pengenalan karena dapat meningkatkan jumlah fitur dari jumlah fitur citra aslinya dan semakin banyak fitur yang dihasilkan akan meningkatkan akurasi klasifikasi [3]. Algoritma *zoning* yang digunakan pada penelitian ini adalah metode *Image Centroid Zone (ICZ)* dan *Zone Centroid Zone (ZCZ)*.

#### 2.6. Metode *Moment Invariant*

*Moment invariant* merupakan fungsi non-linear yang invariant terhadap rotasi, translasi dan skala dalam moment geometri citra [3]. Dalam pemrosesan citra, pengenalan pola dan klasifikasi, metode ini digunakan sebagai fitur.

#### 2.7. Metode *Support Vector Machine (SVM)*

*Support Vector Machine (SVM)* adalah salah satu metode klasifikasi dengan cara mencari *hyperplane* atau garis pemisah *class* yang terbaik pada *input space*. Implementasi SVM diperlukan adanya tahap *training* dan kemudian dilakukan tahap *testing*, oleh karena itu SVM termasuk ke dalam kelas *supervised learning*.

### 3. Hasil dan Pembahasan

#### 3.1. Tahapan *Preprocessing* dan Segmentasi

Proses *preprocessing* dalam penelitian terdapat beberapa tahapan didalamnya yaitu image acquisition dengan mengubah citra input menjadi citra *grayscale* kemudian dilanjutkan dengan melakukan *binarization* untuk mendapatkan citra biner dan dilanjutkan dengan *thinning* atau perangkaan. Penambahan proses *thinning* pada ekstraksi fitur dengan *moment invariant* yang mampu meningkatkan

akurasi klasifikasi[3]. Hasil dari proses *preprocessing* akan lanjut diproses ke dalam proses segmentasi baris. Segmentasi baris dilakukan dengan melakukan pencarian garis segmen menggunakan metode *A\* Path Planning*. Algoritma *A\* Path Planning* menggunakan fungsi heuristik untuk menemukan dan solusi yang tepat untuk mencapai keadaan tujuan dengan menggabungkan biaya jalur aktual dan fungsi heuristik untuk memperkirakan biaya dari *state* awal ke *state* tujuan [1]. Untuk mendapatkan posisi aksara utama dalam masing-masing baris maka perlu dilakukan proyeksi horizontal. Pencarian wilayah aksara utama berguna untuk menentukan area segmentasi masing-masing baris.



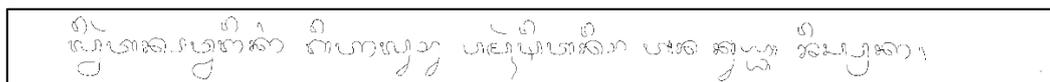
Gambar 3. Pencarian wilayah segmen

Gambar 3 menunjukkan area segmentasi baris yang didapat. Kemudian setelah area segmentasi diketahui maka akan dilanjutkan dengan pencarian garis segmentasi menggunakan metode *A\* Path Planning*.



Gambar 4. Contoh hasil garis segmentasi

Gambar 4 merupakan hasil penentuan garis segmentasi menggunakan metode *A\* Path Planning*. Ketika garis segmentasi telah didapatkan maka dilakukan proses segmentasi dengan mengubah semua piksel di luar wilayah segmen yang diinginkan menjadi piksel putih, hal ini bertujuan untuk meminimalkan munculnya piksel yang tidak diinginkan masuk ke dalam segmentasi. Setelah itu dilanjutkan dengan memotong masing-masing baris lalu menyimpan ke dalam format .png.

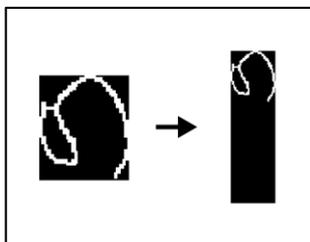


Gambar 5. Contoh hasil segmentasi baris

Gambar 5 merupakan citra baris tersegmentasi yang dihasilkan dari proses segmentasi baris menggunakan metode *A\* Path Planning*.

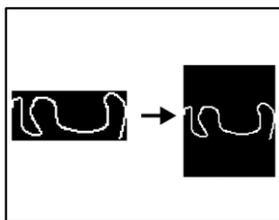
Setelah proses segmentasi baris yang menghasilkan citra baris, maka proses selanjutnya yaitu melakukan segmentasi untuk mendapatkan citra karakter aksara Bali. Segmentasi karakter dilakukan dengan menerapkan algoritma *Component Connected Labeling*. Setelah hasil dari segmentasi menggunakan metode CCL dilakukan seleksi citra label, ketika citra label memiliki ukuran tinggi kurang dari 10 piksel atau memiliki lebar kurang dari 10 piksel maka status label tidak diperlukan (dianggap

noise). Setelah melakukan seleksi citra label maka dilanjutkan dengan mengurutkannya agar berurutan dari atas ke bawah dan kiri ke kanan. Hal ini dilakukan untuk menyesuaikan urutan hasil citra karakter sebelum dikenali dan masuk ke dalam proses pengenalan melalui proses klasifikasi. Hasil dari CCL menghasilkan karakter yang tersambung sama lain, seperti antara pengangge aksara, aksara utama dan gantungan, maka diperlukan pembagian terlebih dahulu untuk memisahkannya. Pembagian dilakukan berpatokan pada wilayah aksara utama, rata-rata tinggi aksara dan wilayah atas dan bawah citra. Setelah itu maka selanjutnya adalah melakukan pembentukan citra berdasar hasil koordinat hasil segmentasi dan melakukan modifikasi dengan menambahkan wilayah citra gelap menyesuaikan dengan posisi citra.



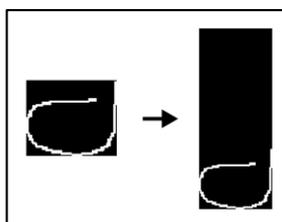
**Gambar 6.** Penambahan Citra Gelap Posisi Karakter di Atas Wilayah Aksara Utama

Gambar 6 merupakan contoh hasil segmentasi aksara *ulu* yang memiliki posisi aksara di atas wilayah aksara utama.



**Gambar 7.** Penambahan Citra Gelap Posisi Karakter di Wilayah Aksara Utama

Gambar 7 merupakan contoh hasil segmentasi aksara *la* yang memiliki posisi aksara berada pada wilayah aksara utama.



**Gambar 8.** Penambahan Citra Gelap Posisi Karakter di Bawah Wilayah Aksara Utama

Gambar 8 merupakan contoh hasil segmentasi aksara *gantungan wa* yang memiliki posisi aksara berada di bawah wilayah aksara utama.

Jika posisi citra merupakan aksara utama maka akan ditambahkan citra gelap di bagian atas dan bawah citra aksara seperti pada Gambar 6, jika posisi citra aksara berada di wilayah atas dari aksara utama maka akan ditambahkan dua citra gelap di bawah citra aksara seperti pada Gambar 7, namun jika posisi citra karakter berapa di bawah wilayah aksara utama maka akan ditambahkan dua citra gelap di atas citra aksara seperti pada Gambar 8.



Gambar 9. Karakter *Ulu*



Gambar 10. Karakter *Gantungan Ga*

Modifikasi hasil segmentasi diatas berguna untuk membedakan beberapa jenis karakter yang bentuknya identik namun dapat dibedakan dari posisi letak karakter tersebut, salah satu contohnya yaitu bentuk karakter *ulu* yang identik dengan *gantungan ga* namun yang membedakan yaitu *ulu* terletak di atas wilayah aksara utama sedangkan *gantungan ga* terletak di bawah aksara utama hal ini dapat dilihat pada Gambar 9 dan 10.

### 3.2. Ekstraksi Fitur

Ekstraksi fitur menerima masukan yaitu berupa citra hasil dari segmentasi karakter. Untuk mencari ciri dari citra karakter pada penelitian ini menggunakan dua buah yaitu metode *Moment Invariant* dan *Zoning (Image Centroid Zone dan Zone Centroid Zone)*. Jumlah fitur yang dihasilkan dengan metode *Moment Invariant* adalah 7 fitur dan metode *Image Centroid Zone dan Zone Centroid Zone* menerapkan pembagian zona  $9 \times 9$  yaitu membagi citra menjadi 81 zona dengan ukuran yang sama. menghasilkan 162 fitur, total jumlah fitur yang dihasilkan pada proses ekstraksi fitur adalah 169 fitur.

### 3.3. Pembentukan Model Pengenalan Aksara

Pembentukan model pengenalan aksara menggunakan 60 data, pada proses ini ditentukan klasifikasi dengan metode *SVM* menggunakan kernel *rbf* dengan nilai  $C = 1000$  dan  $\gamma = 0,0001$ , model tersebut menghasilkan *training score* sebesar 1,000 dan *test score* sebesar 0,903, model yang dihasilkan akan digunakan dalam proses pengenalan aksara untuk mengetahui jenis aksara.

### 3.4. Pengenalan Karakter Aksara Bali

Proses pengenalan karakter aksara bali dilakukan dengan metode *moment invariant* untuk mengekstraksi ciri dari citra karakter tersegmentasi hasil dari proses segmentasi karakter kemudian akan dilakukan klasifikasi menerapkan metode *SVM* dengan melakukan pelatihan dengan data latih dan pengujian akan dilakukan menggunakan data uji dari dataset yang ada dengan mencocokkan ciri dari data uji terhadap ciri yang tersimpan yang merupakan hasil dari pelatihan. Kemudian data yang telah diklasifikasi akan diberikan label menurut kelasnya.

### 3.5. Alih Aksara Bali Berbasis Aturan (*Rule Base*)

Implementasi alih aksara pada penelitian ini dibuat menggunakan rule base yang didalamnya terdapat banyak kondisi menggunakan rangkaian *if-else* yang mengacu pada aturan pembacaan aksara bali. Mengacu kepada aturan pengurutan citra aksara pada implementasi segmentasi aksara. Hasil pengenalan aksara akan ditampung ke dalam sebuah *array*, dan akan dibuat *array* untuk menampung hasil pembacaan *value* dari *array* aksara.

### 3.6. Tampilan Antar Muka Sistem

Tampilan antarmuka sistem dibuat berbasis desktop. Pembuatan tampilan antarmuka sistem pada penelitian ini menggunakan library tkinter dengan bahasa pemrograman *python*. Tampilan antarmuka sistem terlihat pada Gambar 11.



Gambar 11. Tampilan Awal sistem

Gambar 11 merupakan tampilan awal sistem sebelum dilakukan proses pemilihan gambar tulisan tangan dan sebelum proses alih aksara dilakukan. Pengguna dapat memilih data citra aksara bali tulisan tangan dengan menekan "Open" untuk dialihaksarakan.



Gambar 12. Tampilan Sistem

Gambar 12 merupakan tampilan sistem setelah dilakukan proses pemilihan gambar tulisan tangan dan ketika proses alih aksara telah dilakukan. Ketika telah memilih citra maka citra yang dipilih akan ditampilkan oleh sistem, akan muncul tombol "Alih Aksara" yang memiliki fungsi untuk mengalihaksarakan citra yang dipilih, setelah tombol "Alih Aksara" di tekan maka tunggu beberapa waktu hingga tampil huruf latin hasil dari alih aksara oleh sistem.

### 3.7. Pengujian

Skenario selanjutnya yaitu menguji rule base yang dibentuk dengan mencocokkan hasil alih aksara dengan bacaan asli yang pada data yang digunakan saat pengujian. Pengujian selanjutnya yaitu menghitung akurasi alih aksara dari pengujian sistem yang telah dilakukan menggunakan 6 data uji. Penghitungan akurasi dilakukan dengan cara menghitung jumlah huruf hasil alih aksara yang benar susunannya yang merupakan hasil keluaran dari sistem kemudian dibagi dengan jumlah huruf alih aksara yang diharapkan dari masing-masing data. Hasil pengujian alih aksara ditunjukkan pada Tabel 1.

**Tabel 1.** Perhitungan Akurasi Alih Aksara Data *Testing* Sistem

No	Data	Jumlah Huruf Alih Aksara yang Diharapkan	Jumlah Huruf Hasil Alih Aksara oleh Sistem	Akurasi Alih Aksara	Persentase (%)
1	img012_02	184	163	0,885869565	88,587
2	img012_05	183	144	0,786885246	78,689
3	img012_07	181	152	0,839779006	83,978
4	img032_02	185	147	0,794594595	79,459
5	img032_05	184	155	0,842391304	84,239
6	img032_07	183	143	0,781420765	78,142
Rata-rata Akurasi Alih Aksara					82,182

Dari Tabel 1 ditunjukkan bahwa dari data img012\_02 jumlah susunan huruf benar yang dihasilkan oleh sistem yaitu 163 dari 184 huruf yang diharapkan dan menghasilkan akurasi 88,587%, untuk data img012\_05 jumlah susunan huruf benar yang dihasilkan oleh sistem yaitu 144 dari 183 huruf yang diharapkan dan menghasilkan akurasi alih aksara sebesar 78,689%. Selanjutnya untuk data img012\_07 menghasilkan akurasi 83,978% dengan jumlah susunan huruf yang benar yang dihasilkan dari sistem yaitu 152 sedangkan jumlah huruf alih aksara yang diharapkan adalah 182 buah. Kemudian pengujian yang dilakukan dengan data img 032\_02 menghasilkan jumlah susunan huruf yang benar adalah 147 sedangkan jumlah huruf alih aksara yang diharapkan adalah 185 buah huruf sehingga menghasilkan akurasi sebesar 79,459%, data selanjutnya yaitu img032\_05 sistem menghasilkan jumlah susunan aksara yang benar adalah 155 dengan jumlah huruf yang diharapkan adalah 184 buah, dan dari hasil tersebut menghasilkan akurasi sebesar 84,239%. Data terakhir yaitu img032\_07 menghasilkan jumlah huruf dengan susunan yang benar sebanyak 143 dengan jumlah huruf alih aksara yang diharapkan yaitu berjumlah 183 huruf, sehingga menghasilkan akurasi alih aksara 78,142%.

Hasil alih aksara sangat dipengaruhi oleh proses pengurutan aksara dan hasil dari proses pengenalan. Dari pengujian dengan 6 buah data, sistem menghasilkan akurasi alih aksara Bali yaitu dengan rata-rata nilai akurasi sebesar 82,182%, dengan akurasi paling kecil yang dihasilkan pada pengujian dengan data img032\_07 yaitu 78,142 sedangkan hasil akurasi yang paling besar dari pengujian yang telah dilakukan yaitu sebesar 88,587% yaitu hasil pengujian data img012\_02.

#### 4. Kesimpulan

*Rule base* alih aksara Bali dibentuk menggunakan rangkaian kondisi menggunakan *if-else* yang menyesuaikan dengan aturan pembacaan aksara Bali. *Rule base* dirancang mengalihaksarakan susunan aksara yang telah diurutkan saat proses segmentasi karakter dan telah dikenali melewati proses pengenalan aksara. Alih aksara Bali pada tulisan tangan menjadi huruf latin pada penelitian kali ini dihitung dengan membandingkan jumlah huruf latin hasil alih aksara oleh sistem yang benar susunannya dengan jumlah huruf alih aksara yang diharapkan dari masing-masing data, akurasi yang dihasilkan dari perhitungan tersebut menghasilkan rata-rata akurasi sebesar 82,182 % dari 6 data citra yang digunakan dalam pengujian sistem, hasil akurasi yang dihasilkan sangat dipengaruhi oleh urutan aksara segmentasi dan hasil pengenalan aksara.

## Daftar Pustaka

- [1] O. Surinta, M. Holtkamp, F. Karabaa, J. Van Oosten, L. Schomaker, and M. Wiering, "A \* Path Planning for Line Segmentation of Handwritten Documents," no. May, 2014, doi: 10.1109/ICFHR.2014.37.
- [2] I. M. G. Sunarya, M. W. Antara Kesiman, and I. A. P. Purnami, "Segmentasi Citra Tulisan Tangan Aksara Bali Berbasis Proyeksi Vertikal Dan Horisontal," *J. Inform.*, vol. 9, no. 1, pp. 982–992, 2015, doi: 10.26555/jifo.v9i1.a2039.
- [3] 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," vol. 3, no. 2, pp. 91–98, 2019.
- [4] K. Schwenk and F. Huber, "Connected Component Labeling algorithm for very complex and high-resolution images on an FPGA platform," *High-Performance Comput. Remote Sens. V*, vol. 9646, p. 964603, 2015, doi: 10.1117/12.2194101.
- [5] M. Farid, J. Santoso, and E. Setyati, "Handwritten Image Segmentation Carakan Madura Based Projection And Connected Component Labeling," vol. 4, 2020.
- [6] C. P. Yanti, I. G. A. A. M. Aristamy, and I. B. N. Pacima, "Pelabelan Huruf Pada Prasasti Tembaga Menggunakan Thinning Stentiford Dan Connected Component Labelling," *J. Pendidik. Teknol. dan Kegur.*, vol. 17, no. 2, pp. 220–230, 2020.
- [7] R. Aristantya, I. Santoso, and A. A. Zahra, "Identifikasi Tanda Tangan Menggunakan Metode Zoning dan Svm ( Support Vector Machine)," no. March 2018, pp. 1–6, 2018, doi: 10.14710/transient.7.1.174-178.