

## *Aplikasi Pembelajaran Pengenal Aksara Bali Menggunakan Metode Template Matching*

Made Sulatri Dewi<sup>1</sup>, Made Windu Antara Kesiman<sup>2</sup>, I Made Gede Sunarya<sup>3</sup>

Jurusan Pendidikan Teknik Informatika

Universitas Pendidikan Ganesha

Singaraja, Bali

E-mail: cencen\_sweet@yahoo.com<sup>1</sup>, dekndu@yahoo.com<sup>2</sup>, imadegedesunarya@gmail.com<sup>3</sup>

**Abstrak**—Penelitian ini bertujuan untuk merancang dan mengimplementasikan sebuah aplikasi pembelajaran pengenal Aksara Bali untuk memudahkan orang dalam belajar mengenal tulisan Aksara Bali secara lebih efisien dari segi waktu dan tempat. Aplikasi ini dikembangkan dengan menggunakan metode *template matching*. Input dari aplikasi ini berupa tulisan dari user menggunakan *digital pen*, *touchpad* maupun *mouse* dan keluarannya berupa gambar Aksara Bali yang memiliki nilai kesalahan (*error*) paling kecil. Implementasi dari metode *template matching* ini menghasilkan sebuah perangkat lunak yang disebut *Balinese Character Matching*. Aplikasi ini dikembangkan berbasis web sehingga dapat digunakan oleh banyak user tanpa keterbatasan tempat dan waktu. Pada aplikasi ini terdapat dua proses utama yaitu *preprocessing* dan pengenalan dengan metode *template matching*. Proses *preprocessing* terdiri dari *grayscale*, *thresholding* dan normalisasi ukuran. Pada normalisasi ukuran dilakukan proses pencarian batas, *cropping*, penskalaan dan pemberian label 1 dan 0. Uji coba program dilakukan oleh admin (orang yang datanya disimpan pada *database*) dan user (orang yang datanya tidak disimpan pada *database*). Bagi admin yang melakukan uji coba hasil yang diperoleh lebih akurat yaitu dengan rata-rata 95,45%, sedangkan hasil pengujian pada user memperoleh rata-rata sebesar 86,25%.

**Kata kunci**—Aplikasi pembelajaran, Aksara Bali, *Template Matching*, *Balinese Character Matching*, *Preprocessing*

**Abstract**—This research aimed to design and develop an application identifier Balinese character learning that can be used to facilitate for people to recognize text characters Balinese more efficiently in term of time and place. This application is developed by using the *template matching* method. Input of this application is the writing of user using a *digital pen*, *touchpad* or *mouse* and the output is the image of Balinese character that has the smallest error value. Implementation of this *template matching* method produces a software called *Balinese Character*

*Matching*. This application developed based web that can be used by many users without the limitations of space and time. In this application there are two main processes namely *preprocessing* and recognition by *template matching* methods. *Preprocessing* process consists of *grayscale*, *thresholding* and *size normalization*. In the *size normalization* conducted the search process limits, *cropping*, *scaling*, and *labeling* of 1 and 0. Testing program conducted by admin (person whose data is stored in the database) and user (person whose data is not stored in the database). For admin who test the program obtained more accurate result with an average of 95.45%, while the result of testing on the user obtained an average of 86.25%.

**Keywords**—Application of Learning, Balinese Character, *Template Matching*, *Balinese Character Matching*, *Preprocessing*

### I. PENDAHULUAN

Aksara Bali merupakan aksara tradisional masyarakat di Bali yang masih digunakan sampai saat ini. Masyarakat di Bali khususnya sudah mengenal Aksara Bali dari sejak dulu kala, dimana Aksara Bali ini merupakan salah satu identitas kebudayaan masyarakat Bali yang perlu untuk tetap dijaga kelestariannya.

Usaha-usaha pelestarian dan pembinaan bahasa Bali sudah dilakukan sejak zaman dahulu, seperti adanya usaha-usaha penyempurnaan Pedoman Pasang Aksara Bali dan Ejaan Bahasa Bali Huruf Latin. Pengenalan Aksara Bali perlu dilakukan kepada masyarakat luas. Melalui kecanggihan teknologi saat ini yaitu komputer, dapat digunakan untuk mengembangkan sebuah aplikasi dalam menulis Aksara Bali. Upaya pengenalan Aksara Bali melalui komputerisasi ini ditujukan agar masyarakat luas baik yang tinggal di Bali maupun di luar Bali dapat ikut dalam mempelajari penulisan Aksara Bali yang secara tidak langsung ikut untuk melestarikan budaya Aksara Bali.

Aplikasi ini merupakan aplikasi pembelajaran menulis Aksara Bali yang dikembangkan dengan menggunakan metode *template matching*. Terdapat beberapa penelitian terkait dengan penggunaan metode *template matching*, namun dalam penelitian sebelumnya *file* yang diinputkan berupa format \*.jpg ataupun \*.bmp. Misalnya penelitian yang dilakukan oleh Suryo Hartanto, Aris Sugiharto, Sukmawati Nur Endah dihasilkan sebuah aplikasi *optical character recognition* dengan menggunakan algoritma *template matching correlation* untuk pengenalan karakter huruf cetak dengan rata-rata tingkat keberhasilan pengenalan sebesar 92,90%[1]. Perbedaannya dengan penelitian yang telah dilakukan sebelumnya adalah *user* dapat menulis secara langsung karakter yang diinginkan.

Berdasarkan latar belakang di atas, maka penulis memiliki inisiatif untuk mengembangkan aplikasi pembelajaran pengenalan Aksara Bali menggunakan metode *template matching*.

## II. KAJIAN TEORI

### A. Aksara Bali

Menurut Gautama (2006:32), aksara adalah ciri-ciri atau suatu gambaran suara yang diciptakan oleh manusia [2]. Aksara Bali adalah huruf suku kata, tanpa mendapat *pengangge* suara sudah dapat berfungsi sebagai suku kata [3]. Pembagian Aksara Bali ditunjukkan pada Tabel 1.

Tabel 1. Pembagian Aksara Bali Berdasarkan Bentuk dan Fungsinya [4]

Pembagian Aksara Bali Berdasarkan bentuk dan fungsinya		
Aksara Biasa	Aksara Wreastra	Aksara sehari-hari (berjumlah 18 aksara)
	Aksara Swalalita	Aksara kesusastraan kawi (berjumlah 47 aksara)
Aksara Suci	Aksara Wijaksara/Bijaksara	Aksara Swalalita dan aksara amsa
	Aksara Modre	Aksara lukisan magis

### B. Pengolahan Citra Digital

Citra menurut kamus Webster, adalah suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda. Citra merupakan istilah lain untuk gambar sebagai salah satu komponen multimedia yang memegang peranan penting dalam informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks yaitu citra kaya dalam informasi.

Suatu citra dapat didefinisikan sebagai fungsi  $f(x,y)$  berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo  $f$  di titik koordinat  $(x,y)$  dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut, jika nilai x, y, dan nilai amplitudo  $f$  secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital [5].

### C. Preprocessing Citra Digital

*Preprocessing* adalah tahap pertama yang harus dilakukan pada proses pencocokan *character* sebelum menuju ke tahap pengenalan *template matching*. Tahap ini sangat penting untuk menentukan keberhasilan suatu proses pengenalan pola.

- 1) Konversi citra RGB (*Red,Green,Blue*) menjadi citra *Grayscale*

Proses konversi citra RGB (*Red,Green,Blue*) menjadi citra *grayscale* ini dilakukan untuk memperkecil ukuran data citra yang akan diolah sehingga waktu komputasi akan berjalan lebih cepat. Proses *grayscale* ini dilakukan sebelum dilakukan *thresholding* untuk mendapatkan citra biner yang akan diolah. Salah satu cara yang paling mudah untuk memperoleh nilai citra *grayscale* dari citra RGB, dilakukan dengan menghitung rata-rata nilai intensitas RGB dari setiap piksel penyusun citra tersebut

- 2) *Thresholding*

*Thresholding* adalah proses mengubah citra berderajat keabuan (*grayscale*) menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk obyek dan *background* dari citra secara jelas. Citra hasil *thresholding* ini selanjutnya digunakan untuk proses pengenalan obyek serta ekstraksi fitur.

Hasil *thresholding* merupakan citra biner, dimana semua piksel dengan tingkat keabu-abuan

yang lebih tinggi daripada nilai *threshold* akan diklasifikasikan sebagai objek dengan nilai *pixel* 0 dan sisa piksel diklasifikasikan sebagai *background* dengan nilai *pixel* 255. *Thresholding* yang dilakukan mengacu pada formula sebagai berikut.

$$I_{Bin}(x,y) = \begin{cases} 0 & I_{BW}(x,y) > T \\ 255 & I_{BW}(x,y) \leq T \end{cases} \quad (1)$$

### 3) Normalisasi

Normalisasi merupakan salah satu tahap dalam *preprocessing* citra yang dilakukan sebelum masuk ke proses pengenalan. Proses ini dilakukan dengan tujuan untuk menyesuaikan ukuran data citra masukan dengan data citra pada basis data. Proses normalisasi disesuaikan dengan kebutuhan pada proses pengenalan yang digunakan. Salah satu proses normalisasi yang paling sederhana dalam pengenalan pola adalah normalisasi ukuran citra.

#### a. Proses Pencarian Batas (*Search Object*)

Setelah karakter di *threshold*, maka karakter akan dicari batas atas, bawah, kiri, dan kanannya. Pencarian batas digunakan untuk mengetahui batas-batas karakter yang menentukan bagian citra yang akan diolah. Proses pencarian batas ini dilakukan dengan mencari nilai *pixel* yang berwarna hitam pada setiap batas atas, kanan, bawah dan kiri dari karakter.

#### b. *Cropping* Karakter

*Cropping* karakter adalah proses **pemotongan citra** pada koordinat tertentu pada area citra. Sehingga akan membentuk bangun segi empat yang mana tiap-tiap *pixel* yang ada pada area koordinat tertentu akan disimpan dalam citra yang baru.

#### c. Proses Penskalaan

Penskalaan adalah mengubah ukuran citra, baik memperbesar ataupun memperkecil ukuran citra dengan mempertahankan proporsionalitasnya.

Posisi tiap *pixel* hasil penskalaan dapat dinyatakan dengan menggunakan persamaan 2 dan persamaan 3 sebagai berikut:

$$X1 = X/dx \quad (2)$$

$$Y1 = Y/dy \quad (3)$$

X1 dan Y1 merupakan koordinat hasil penskalaan. Sedangkan dx adalah ratio skala arah mendatar dan skala dy adalah ratio skala arah tegak.

#### d. Pemberian label karakter (1 dan 0)

Pemberian label ini dilakukan dengan tujuan untuk membedakan antara objek karakter dan *background*. Setelah terbentuk citra biner yang telah mengalami penskalaan sesuai dengan ukuran citra di *database* selanjutnya nilai *pixel* yang berwarna hitam akan diberi label 1, sedangkan nilai *pixel* yang berwarna putih akan diberi label 0.

### D. Metode *Template Matching*

Algoritma *template matching* merupakan metode yang sederhana dan banyak digunakan untuk mengenali pola Metode *template matching* ini merupakan metode yang sederhana. Suatu citra masukan yang mengandung *template* tertentu dibandingkan dengan *template* pada basis data. *Template* ditempatkan pada pusat bagian citra yang akan dibandingkan dan dihitung seberapa banyak titik yang paling sesuai dengan *template*. Langkah ini diulangi terhadap keseluruhan citra masukan yang akan dibandingkan. Nilai kesesuaian titik yang paling mendekati atau *error* minimum yang paling kecil antara citra masukan dan citra *template* menandakan bahwa *template* tersebut merupakan citra *template* yang paling sesuai dengan citra masukan.

Tingkat kesesuaian antara citra masukan dan citra *template* bisa dihitung berdasarkan nilai *error* terkecil dengan menggunakan persamaan 1.

$$\text{Min } e = \sum (I_{x,y} - T_{x,y})^2 \quad (4)$$

I adalah pola *pixel* citra masukan yang akan dibandingkan. T adalah pola *pixel* citra *template*. *Template* dengan nilai *error* paling kecil adalah *template* yang paling sesuai dengan citra masukan yang akan dibandingkan. Contoh berikut ini adalah penerapan pengenalan karakter dengan menggunakan metode *templatematching* yang menggunakan persamaan (4).

1. Tabel 2 merupakan contoh deretan angka biner citra *template* yang disimpan dalam basis data. Dalam tabel tersebut terdapat empat buah citra *template* yaitu *template A*, *template B*, *template C* dan *template D*.

Tabel 2. Contoh Deretan Angka Biner Citra Karakter Template[6]

Karakter	Deretan angka biner citra template
A	0 0 1 1 1 0 0 0 0
B	1 1 1 0 0 0 1 1 0
C	1 0 1 0 0 0 1 1 1
D	0 1 1 1 0 1 0 0 1

2. Deretan berikut ini adalah deretan angka biner citra karakter masukan yang ditunjukkan pada Tabel 3.

Tabel 3. Deretan Angka Biner Citra Masukan

Citra Masukan	0 0 0 1 0 1 0 1 0
---------------	-------------------

3. Tabel 4 adalah perhitungan *error* dengan menggunakan persamaan (4):

Tabel 4. Perhitungan Nilai Error [6]

Karakter	Perhitungan nilai error
A	$(0-0)^2 + (0-0)^2 + (0-1)^2 + (1-1)^2 + (0-1)^2 + (1-0)^2 + (0-0)^2 + (1-0)^2 + (0-0)^2 = 4$
B	$(0-1)^2 + (0-1)^2 + (0-1)^2 + (1-0)^2 + (0-0)^2 + (1-0)^2 + (0-1)^2 + (1-1)^2 + (0-0)^2 = 6$
C	$(0-1)^2 + (0-0)^2 + (0-1)^2 + (1-0)^2 + (0-0)^2 + (1-0)^2 + (0-1)^2 + (1-1)^2 + (0-1)^2 = 6$
D	$(0-0)^2 + (0-1)^2 + (0-1)^2 + (1-1)^2 + (0-0)^2 + (1-0)^2 + (0-0)^2 + (1-0)^2 + (0-1)^2 = 5$

Jika nilai *error* masing-masing *template* sudah diketahui, maka akan dicari nilai *error* terkecil. Nilai *error* terkecil adalah *template* yang paling sesuai dengan karakter. Nilai terkecil dari hasil perhitungan pada tabel 4 adalah 4. Maka citra *template* yang paling sesuai dengan citra masukan adalah citra karakter A.

### III METODOLOGI

#### A. Analisis Perangkat Lunak

Pada akhir dari penelitian ini, akan dibangun sebuah perangkat lunak berbasis web. Perangkat lunak yang dibangun mampu mengenali Aksara Bali yang telah ditulis oleh pengguna dalam sebuah canvas yang menggunakan *digital pen*, *touchpad* ataupun *mouse* kemudian membandingkan dengan *template* yang telah disimpan di dalam basis data.

#### 1. Kebutuhan Perangkat Lunak

Berdasarkan analisis yang dilakukan terhadap pengembangan aplikasi pembelajaran pengenalan Aksara Bali dengan metode *template matching* terdapat beberapa proses yang dapat diimplementasikan, yaitu :

1. Menulis objek/karakter dengan menggunakan *digital pen*, *touchpad* atau *mouse*.

2. Melakukan konversi citra RGB (*Red, Green, Blue*) menjadi citra *Grayscale* sebelum nantinya dilakukan proses *thresholding* untuk memperoleh citra biner yang akan diproses dengan menggunakan metode *template matching*.

3. Melakukan proses perubahan citra keabuan (*grayscale*) menjadi citra hitam putih atau biner pada citra yang dinamakan dengan proses binerisasi atau *thresholding*.

4. Melakukan proses pencarian batas objek yang akan diolah nantinya sebelum dilakukan proses *cropping*.

5. Melakukan proses *cropping* objek/karakter yang nantinya hanya objek/karakter itu saja yang akan diolah dengan metode *template matching*.

6. Melakukan penskalaan ukuran citra untuk menyamakan ukuran antara citra masukan yang ditulis oleh *user* dengan citra *template* yang disimpan di dalam *database* jika ukuran citra masukan memiliki tinggi lebih atau kurang dari 50 *pixel* dan lebar lebih atau kurang dari 50 *pixel*.

7. Melakukan pemberian label karakter dan menampilkan fitur karakter dimana karakter objek akan direpresentasikan dengan nilai 1 untuk objek yang berwarna hitam sedangkan background yang berwarna putih akan direpresentasikan dengan nilai 0.

8. Melakukan penyimpanan *template* Aksara Bali beserta nilai fitur citra karakter pada *database* dengan *preprocessing* yang sama dengan citra masukan.

9. Melakukan pengenalan karakter Aksara Bali antara citra masukan dengan citra *template* dengan mencari nilai *error* minimum yang dikenal dengan metode *template matching*.

#### 2. Tujuan Pengembangan Perangkat Lunak

Aplikasi *Balinese Character Matching* ini diharapkan mampu memenuhi proses-proses sebagai berikut.

a. Mampu menulis objek/karakter dengan menggunakan *digital pen*, *touchpad* atau *mouse*.

b. Mampu melakukan konversi citra RGB (*Red, Green, Blue*) menjadi citra *Grayscale* sebelum nantinya dilakukan proses *thresholding* untuk memperoleh citra biner



- yang akan diproses dengan menggunakan metode *template matching*.
- c. Mampu melakukan proses perubahan citra keabuan (*grayscale*) menjadi citra hitam putih atau biner pada citra yang dinamakan dengan proses binerisasi atau *thresholding*.
  - d. Mampu melakukan proses pencarian batas objek yang akan diolah nantinya sebelum dilakukan proses *cropping*.
  - e. Mampu melakukan proses *cropping* objek/karakter yang nantinya hanya objek/karakter itu saja yang akan diolah dengan metode *template matching*.
  - f. Mampu melakukan penskalaan ukuran citra untuk menyamakan ukuran antara citra masukan yang ditulis oleh *user* dengan citra *template* yang disimpan di dalam *database* jika ukuran citra masukan memiliki tinggi lebih atau kurang dari 50 *pixel* dan lebar lebih atau kurang dari 50 *pixel*.
  - g. Mampu melakukan pemberian label karakter dan menampilkan fitur karakter dimana karakter objek akan direpresentasikan dengan nilai 1 untuk objek yang berwarna hitam sedangkan background yang berwarna putih akan direpresentasikan dengan nilai 0.
  - h. Mampu melakukan penyimpanan *template* Aksara Bali beserta nilai fitur citra karakter pada *database* dengan *preprocessing* yang sama dengan citra masukan.
  - i. Mampu melakukan pengenalan karakter Aksara Bali antara citra masukan dengan citra *template* dengan mencari nilai error minimum yang dikenal dengan metode *template matching*.
3. *Masukan dan Keluaran Perangkat Lunak*  
 Masukan untuk perangkat lunak *Balinese Character Matching* yaitu sebagai berikut.
- a. *Masukan oleh user*  
 Tulisan Aksara Bali yang ditulis oleh *user* dengan menggunakan *digital pen*, *touchpad* atau *mouse*. *User* dapat memilih jenis aksara yang ingin dibuat. Selain itu *user* juga dapat memilih jenis warna yang diinginkan untuk menulis aksara pada *canvas* yang telah disediakan. Adapun citra yang dapat diolah adalah citra RGB dengan kedalaman warna yaitu 24 bit.
  - b. *Masukan pengetahuan oleh admin*
    - Penambahan data admin. Data admin yang ditambahkan meliputi id, nama, *username* dan password.
    - Penambahan jenis Aksara Bali. Contoh jenis aksara ini meliputi aksara wianjana, aksara suara, aksara gantungan, aksara tengenan, angka.
    - Penambahan *file*/gambar Aksara Bali yang dimasukkan oleh admin berupa *file* Aksara Bali bertipe bitmap (\*.bmp), JPEG (\*.jpeg), PNG (\*.png). Aksara ini nantinya akan ditambahkan beberapa jenis *template* yang dicocokkan dengan Aksara Bali yang ditulis oleh *user*.
    - Penambahan *template* Aksara Bali (citra) pada tiap *file* aksara yang sudah ada. Masing-masing Aksara Bali mempunyai lebih dari satu *template*.

Keluaran dari perangkat lunak aplikasi *Balinese Character Matching* yaitu :

    - a. *Keluaran yang diterima oleh user* :
      - Gambar Aksara Bali yang paling mendekati dengan tulisan yang *diinput* oleh *user*.
      - Persentase nilai perhitungan error minimum dan persentase nilai kebenaran dengan Aksara Bali yang sebenarnya.
      - Fitur citra Aksara Bali berupa citra biner hasil dari proses *thresholding* yang telah direpresentasikan dengan nilai 1 (untuk *pixel* yang berwarna hitam) dan nilai 0 (untuk *pixel* yang berwarna putih).
    - b. *Keluaran yang diterima oleh admin* :
      - Informasi mengenai data admin.
      - Informasi mengenai data jenis aksara.
      - Informasi mengenai *file*/gambar Aksara Bali yang telah disimpan.
      - Informasi mengenai *template* Aksara Bali yang telah dibuat dan disimpan pada *database*.
      - Informasi mengenai fitur citra Aksara Bali yang telah mengalami proses *preprocessing*.

### B. Perancangan Perangkat Lunak

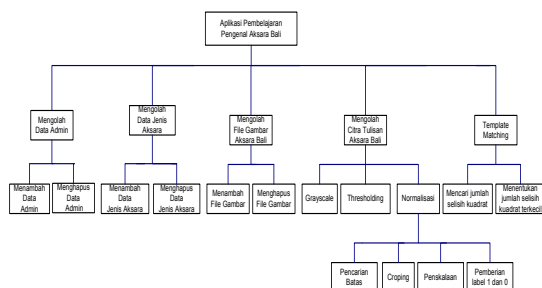
Tahapan berikutnya setelah melakukan analisis perangkat lunak adalah melakukan perancangan perangkat lunak. Rancangan yang dibuat bersifat *userfriendly* agar pengguna merasa nyaman dan mudah dalam menggunakan aplikasi ini.

#### 1) Batasan Perancangan Perangkat Lunak

Batasan perancangan aplikasi *Balinese Character Matching* yaitu *user* hanya mampu memberikan *inputan* berupa tulisan Aksara Bali yang ditulis menggunakan *digital pen*, *touchpad* atau *mouse* pada *canvas* yang telah disediakan. *User* tidak dapat memberikan *inputan* berupa *file* citra ataupun hasil scan gambar tertentu.

#### 2) Perancangan Arsitektur Perangkat Lunak

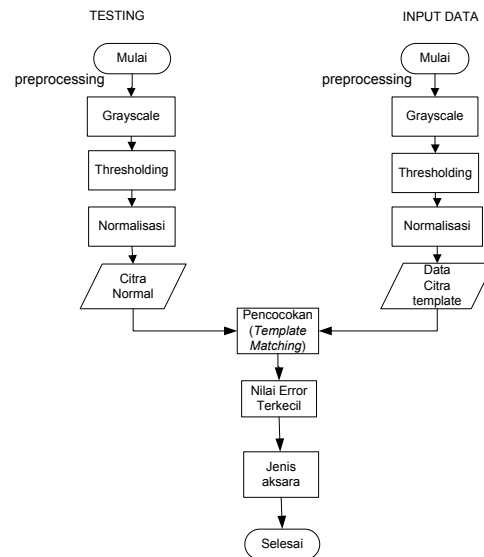
Perancangan arsitektur perangkat lunak menggambarkan bagian-bagian modul, struktur ketergantungan antar modul, dan hubungan antar modul dari perangkat lunak yang dibangun. *Structure chart* dari aplikasi *Balinese Character Matching* ditunjukkan pada Gambar 1.



Gambar 1. Structure Chart Aplikasi *Balinese Character Matching*

#### 3) Diagram Alir Aplikasi Pembelajaran Pengenal Aksara Bali

Dalam diagram alir ini akan dijelaskan mengenai proses-proses yang terjadi dalam aplikasi pembelajaran pengenal aksara bali dengan menggunakan metode *template matching*. Diagram alir aplikasi pembelajaran pengenal Aksara Bali ditunjukkan pada Gambar 2.



Gambar 2. Diagram Alir Aplikasi Pembelajaran Pengenal Aksara Bali

Pada Gambar 2 diagram alir aplikasi pembelajaran pengenal Aksara Bali dibagi menjadi dua bagian yaitu bagian *testing* dan *input data*. Proses pengenalan karakter Aksara Bali dengan metode *template matching* ini diawali dari proses *penginputan* data terlebih dahulu oleh admin. Data yang *diinputkan* berupa *file* gambar Aksara Bali bertipe bitmap (\*.bmp), JPEG (\*.jpeg), PNG (\*.png). Masing-masing *file* gambar Aksara Bali yang telah *diinputkan* akan ditambahkan beberapa buah *template*. Admin menambahkan *template* dengan cara menulis Aksara Bali pada *canvas* yang telah disediakan. Selanjutnya citra Aksara Bali ini akan mengalami proses *preprocessing* yang terdiri dari proses *grayscale*, *thresholding* dan normalisasi ukuran. Data yang telah mengalami *preprocessing* dalam *penginputan* data ini akan menghasilkan data citra *template* yang disimpan dalam basis data. Proses pengenalan karakter (*testing*) dimulai saat *user* mulai menggambar karakter yang diinginkan dengan menggunakan *digital pen*, *touchpad* ataupun *mouse* sesuai dengan keinginan *user*. Karakter yang digambar tersebut nantinya juga akan mengalami proses *preprocessing* yang sama dengan *preprocessing* pada *penginputan* data oleh admin. Melalui proses *testing* ini akan dihasilkan data citra normal yang selanjutnya akan dicocokkan dengan data citra *template* pada basis data. Proses pencocokan ini menggunakan metode *template matching*. Masing-masing *pixel* pada citra normal dan citra *template* yang bersesuaian

akan dicari selisih kuadratnya kemudian akan dicari total selisih yang paling kecil. Proses ini dinamakan pencarian nilai *error* terkecil. Setelah itu sistem akan menemukan jenis aksara mana yang paling sesuai dengan data yang telah diinputkan oleh *user* dengan nilai *error* terkecil. Hasil akhir dari aplikasi ini akan menampilkan gambar Aksara Bali yang memiliki nilai *error* terkecil, nama Aksara Bali, persentase nilai *error* serta persentase nilai kebenaran dengan aksara yang sebenarnya.

#### 4. Perancangan Struktur Data Perangkat Lunak

Perancangan struktur data perangkat lunak merupakan tahap pendefinisian dari kebutuhan-kebutuhan fungsional dalam suatu tahap pengembangan sistem. Pada aplikasi ini menggunakan *database* yang terdiri dari empat buah tabel yaitu:

##### 1) Tbadmin

Tbadmin merupakan tabel tempat disimpannya data-data admin sebagai pengguna yang memasukkan pengetahuan. Rancangan tbadmin ditunjukkan pada Tabel 5.

Tabel 5. tbadmin

Nama	Tipe Data	Keterangan
Id	Int (11)	Id admin (primary key)
Nama	Text	Nama admin
Username	Varchar (30)	Username
password	Varchar (30)	Password

##### 2) Tbjenisaksara

Tbjenisaksara merupakan tabel tempat disimpannya data-data jenis aksara seperti id dan jenis. Rancangan tbjenisaksara ditunjukkan pada Tabel 6.

Tabel 6. tbjenisaksara

Nama	Tipe Data	Keterangan
Id	Int (11)	Id jenis aksara (primary key)
jenis	Varchar (60)	Jenis aksara

##### 3) Tbaksara

Tbaksara merupakan tabel tempat disimpannya data-data aksara yang diupload oleh admin. Rancangan Tbaksara ditunjukkan pada Tabel 7.

Tabel 7. tbaksara

Nama	Tipe Data	Keterangan
Id	Int (11)	Id aksara (primary key)
huruf	Varchar (60)	Huruf latin
jenis	Int (11)	Jenis aksara
file	Varchar (60)	Gambar aksara

##### 4) Tbfituraksara

Tbfituraksara merupakan tabel tempat disimpannya data *template* aksara. Pada tabel ini disimpan fitur citra yang merupakan nilai *pixel* dari masing-masing *template* yang sudah menjadi citra biner. Rancangan tbfituraksara ditunjukkan pada Tabel 8.

Tabel 8. tbfituraksara

Nama	Tipe Data	Keterangan
Id	Int (11)	Id aksara (primary key)
idaksara	Int (11)	Id <i>template</i> aksara
file	varchar (60)	File <i>template</i> aksara
fitur	text	Fitur <i>template</i> aksara

## IV. PEMBAHASAN

### A. Implementasi Perangkat Lunak

Pada implementasi perangkat lunak *Balinese Character Matching* ini akan dijelaskan mengenai beberapa hal yaitu lingkungan implementasi perangkat lunak, implementasi arsitektur perangkat lunak, implementasi struktur data perangkat lunak serta implementasi layar antarmuka dari perangkat lunak yang dibangun.

#### 1) Lingkungan Implementasi Perangkat Lunak

##### a. Lingkungan Perangkat Keras .

Perangkat keras yang digunakan adalah laptop dengan spesifikasi sebagai berikut:

- 1) Monitor 12 inch
- 2) Processor AMD Athlon™ Neo Processor 1.6 GHz
- 3) RAM 2 GB
- 4) Hard Disk 300 GB
- 5) Dilengkapi dengan alat *input* dan *output*

##### b. Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun sistem ini adalah sebagai berikut:

- 1) Sistem operasi : Ubuntu-12.04
- 2) Software aplikasi : PHP
- 3) Software *database* : MySQL
- 4) Software Pendukung :

- Bluefish
- XAMPP
- Mozilla Firefox 12.0

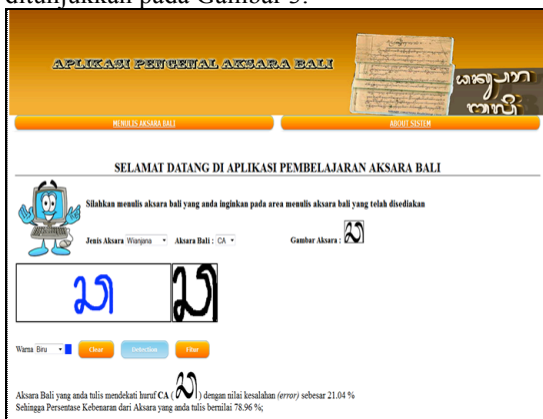
## 2) Batasan Implementasi Perangkat Lunak

Batasan implementasi yang terdapat pada perangkat lunak *Balinese Character Matching* ini yaitu hanya mampu mengolah citra dengan *inputan* tulisan dengan menggunakan *digital pen*, *touchpad* ataupun *mouse* dan tidak mampu mengolah citra *inputan* berupa hasil scan atau gambar yang diupload.

## 3) Implementasi Layar Antarmuka perangkat Lunak

### a. Menu Halaman Utama

Implementasi menu halaman utama ditunjukkan pada Gambar 3.



Gambar 3. Menu Halaman Utama

### b. Menu Tambah Aksara

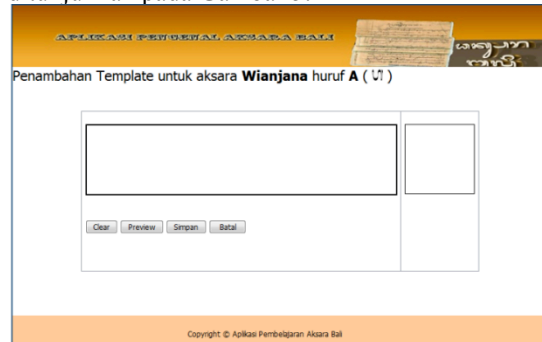
Implementasi menu tambah aksara ditunjukkan pada Gambar 4.



Gambar 4. Menu Tambah Aksara

### c. Menu Tambah Template

Implementasi menu tambah *template* ditunjukkan pada Gambar 5.



Gambar 5. Menu Tambah Template

## 4) Pengujian Perangkat Lunak

### a. Tujuan Pengujian Perangkat Lunak

Tujuan pengujian perangkat lunak adalah untuk mengetahui mampu tidaknya pengguna sistem dalam menjalankan aplikasi *Balinese Character Matching* dan sejauh mana keakuratan sistem dapat menjalankan fungsi-fungsi sistem yang terdapat pada aplikasi *Balinese Character Matching*.

1. Untuk mengetahui keakuratan berjalannya fungsi-fungsi manipulasi data pendukung seperti data admin, data jenis aksara dan data aksara yang digunakan.
2. Untuk mengetahui kebenaran dari proses preprocessing yang dilakukan beserta fitur citra yang dihasilkan dari preprocessing yang dilakukan pada masing-masing aksara yang diujicobakan.
3. Untuk mengetahui kebenaran proses dalam memasukkan *template* Aksara Bali.
4. Untuk mengetahui kebenaran hasil perhitungan nilai error yang dihasilkan dengan metode *template matching*.
5. Untuk mengetahui kebenaran dari *output* sistem yang dihasilkan yaitu antara *inputan user* dengan kemiripan *output* Aksara Bali yang ditampilkan.

### b. Pelaksanaan Pengujian Perangkat Lunak

Berdasarkan perancangan pengujian perangkat lunak, maka pengujian aplikasi *Balinese Character Matching* dilakukan oleh: 1. Pengembang untuk pengujian fungsionalitas



proses-proses yang terjadi pada perangkat lunak; 2. Dua orang admin sebagai orang yang menambahkan *template* Aksara Bali untuk disimpan pada *database*. Pengujian dilakukan sesuai dengan kasus uji yang telah dirancang sebelumnya dengan menggunakan angket pengisian *template*; 3. Lima orang pengguna dimana dua diantaranya merupakan admin yang sudah menambahkan *template* terlebih dahulu pada *database*. pengujian dilakukan sesuai dengan kasus uji yang telah dirancang sebelumnya dengan menggunakan sebuah angket kesesuaian *input* dan *output* Aksara Bali yang dihasilkan. Hasil dari angket tersebut dapat dilihat pada Tabel 9.

Tabel 9. Hasil Uji Coba Aplikasi *Balinese Character Matching*

Pengguna ke -	Jumlah Aksara Bali yang dicocokkan		Persentase keakuratan	Keterangan
	Sam a	Berbeda		
1	158	7	95.75 %	Admin
2	157	8	95,15 %	Admin
3	148	17	89.69 %	Pengguna
4	141	24	85.45 %	Pengguna
5	138	27	83.63 %	Pengguna

Pada Tabel 9 merupakan hasil uji coba pencocokan aksara bali dengan menggunakan metode *template matching*. Pada uji coba ini menggunakan 55 buah karakter aksara bali yang terdiri dari 18 buah aksara wianjana, 18 buah gantungan aksara wianjana, 10 buah angka aksara bali, 5 buah aksara suara, dan 4 buah tengenan. Masing-masing aksara akan dicoba sebanyak tiga kali oleh masing-masing pengguna untuk memperoleh hasil yang lebih akurat. Adapun hasil aksara bali yang dicocokkan secara lebih detail dapat dilihat pada Tabel 10.

Tabel 10. Hasil Uji Coba Aplikasi *Balinese Character Matching* untuk Aksara Bali yang Berbeda

Pengguna ke -	Aksara Bali yang Berbeda	Keterangan
1	Aksara wianjana (sa, wa, la, ga), angka 1, aksara suara e (taleng), aksara suara u (suku).	admin
2	Aksara wianjana (wa, la), gantungan ba, angka 1, angka 2, angka 7, aksara e (taleng), tengenan adeg-adeg.	admin
3	Aksara wianjana (a, ra, da, ta), gantungan na, gantungan ca, gantungan wa, gantungan la, gantungan ya, angka 1, angka 2, angka 5, aksara suara tedong.	Pengguna (Pada aksara ra, ta dan gantungan na terdapat dua kali penulisan aksara yang berbeda)
4	Aksara wianjana (a, na, ca, ta, wa, ga, ya) gantungan na, gantungan ca, gantungan ka, gantungan da, gantungan ta, gantungan wa, gantungan ya, angka 7, angka 9, aksara taleng, aksara ulu, tengenan surang, tengenan bisah, tengenan adeg-adeg.	Pengguna (Pada aksara wa, ulu dan tengenan adeg-adeg terdapat dua kali penulisan aksara yang berbeda)
5	Aksara wianjana (na, ca, ra, da, ta, la, ma, ga, ba), gantungan na, gantungan ka, gantungan ta, gantungan ya, angka 1, angka 2, angka 7, aksara suara ulu, aksara suara tedong.	Pengguna (aksara da, ba dan angka 1 terdapat dua kali penulisan aksara yang berbeda sedangkan pada aksara ta, la dan angka 2 terdapat tiga kali penulisan aksara yang berbeda)

### c. Evaluasi Hasil Pengujian perangkat Lunak

Berdasarkan hasil pengujian fungsional pada proses-proses yang terjadi pada perangkat lunak diketahui bahwa fungsi-fungsi manipulasi data pendukung seperti data admin, data jenis aksara dan data aksara sudah berjalan dengan baik. Begitu pula halnya dengan proses preprocessing sudah berjalan dengan benar. Hal ini dapat dilihat dari hasil yang ditampilkan, baik itu proses *grayscale*, *thresholding*, dan normalisasi serta pemberian label 1 dan 0 untuk disimpan pada *database*. Selain itu dari segi hasil perhitungan yang diperoleh terdapat hasil yang sama antara perhitungan manual dan perhitungan dengan sistem yang ada. Hal ini membuktikan bahwa metode *template matching* yang digunakan sudah sesuai dengan konsep yang ada.

Berdasarkan pengujian yang dilakukan kepada admin dalam mengisi *template*, dari segi kemudahan menulis *template* lebih mudah untuk menggunakan alat pendukung yaitu *digital pen* daripada menggunakan *touchpad* ataupun *mouse*. Selain itu dari segi kemudahan menggunakan menu yang ada, kualitas *template* yang disimpan

dan kecepatan proses penyimpanan *template* sudah berjalan dengan baik.

Berdasarkan pengujian yang dilakukan untuk menguji kesesuaian hasil *input* dan *output* Aksara Bali diperoleh keakuratan uji coba pada admin (orang yang data *templat*nya disimpan pada *database*) menghasilkan nilai yang lebih besar dibandingkan *user* (orang yang data *templat*nya belum disimpan pada *database*). Pada admin diperoleh rata-rata keakuratan sistem sebesar 95.45 % sedangkan pada *user* sebesar 86.25 %.

## V. SIMPULAN

Berdasarkan penelitian yang telah dilakukan yaitu “Aplikasi Pembelajaran Pengenal Aksara Bali dengan Menggunakan Metode *Template Matching*” diperoleh simpulan sebagai berikut:

1. Aplikasi pembelajaran pengenal Aksara Bali ini dikembangkan dengan metode *template matching* dengan *input* berupa tulisan Aksara Bali dari *user* menggunakan *digital pen*, *touchpad* ataupun *mouse*. Aplikasi ini sudah berjalan dengan baik sebagai sarana untuk mempelajari Aksara Bali. Aplikasi ini diimplementasikan berbasis web sehingga dapat digunakan oleh banyak *user* tanpa keterbatasan tempat dan waktu. Uji coba aplikasi ini memperoleh hasil yang lebih akurat pada admin (orang yang data *templat*nya disimpan pada *database*) dengan rata-rata 95.45% dibandingkan *user* (orang yang data *templat*nya tidak disimpan pada *database*) dengan rata-rata 86.25%.
2. Dalam pengembangan aplikasi *Balinese Character Matching* selanjutnya dapat ditambahkan *template* yang lebih banyak pada masing-masing aksara yang disimpan dalam *database* untuk memperoleh hasil yang lebih optimal. Pada *interface* untuk mempermudah *user* dalam menggunakan aplikasi ini dapat juga ditambahkan dengan penggunaan *flash* setiap gambar Aksara Bali tentang cara menulis Aksara Bali yang benar ketika *user* memilih jenis Aksara Bali yang ingin ditulis. Dalam pengembangan yang lebih lanjut bagi pembaca yang tertarik dapat mengembangkan aplikasi ini dengan berbasis lainnya seperti *mobile* ataupun *android*.

## REFERENSI

- [1] Hartanto Suryo, Aris Sugiharto, Sukmawati Nur Endah. 2012. “Optical Character Recognition Menggunakan Algoritma *Template Matching Correlation*”. Jurnal of Informatics and Technology, Vol 1, No.1, Tahun 2012 (hlm11-20)
- [2] Gautama, Wayan Budha. 2006. Tata Sukerta Basa Bali. Denpasar: CV. Kayumasagung
- [3] Medera, dkk. 2002. Pedoman Pasang Aksara Bali. Denpasar: Dinas Kebudayaan Propinsi Bali
- [4] Nala, Ngurah. 2006. Aksara Bali dalam Usada. Surabaya: Paramita Surabaya.
- [5] Putra, Darma. 2010. Pengolahan Citra Digital. Yogyakarta: Andi
- [6] Bahri, Raden Sofyan dan Irfan Maliki. 2012. “Perbandingan Algoritma *Template Matching* Dan *Feature Extraction* Pada *Optical Character Recognition* “. Jurnal Komputer dan Informatika (KOMPUTA) Edisi 1, Volume 1. (hlm. 29-35)