BAB III
ANALISIS DAN PERANCANGAN

Tahap analisis dan perancangan merupakan tahapan identifikasi masalah serta kebutuhan dari suatu sistem yang sedang berjalan, sehingga masalah yang telah di rincikan pada tahap sebelumnya dapat terselesaikan dan selanjutnya melakukan rancangan dari sistem yang akan dibuat.

3.1 Analisis Masalah

Aplikasi yang akan di implementasikan ini merupakan aplikasi yang dapat membandingkan algoritma *Template Matching* dengan algoritma *Feature Extraction* pada studi kasus pengenalan aksara sunda dengan inputan aksara sunda dalam format gambar. aplikasi ini menggunakan *offline character recognition* dimana data *template* (data latih) dapat diakses secara offline. Gambar inputan aksara sunda yang akan dikenali dapat bersumber dari *capture* kamera atau dari gambar galeri berformat .PNG atau .JPG. kualitas gambar inputan berpengaruh pada tingkat akurasi pembacaan dari algoritma, untuk itu diperlukannya suatu gambar dengan *noise* yang sedikit agar akurasinya tinggi.

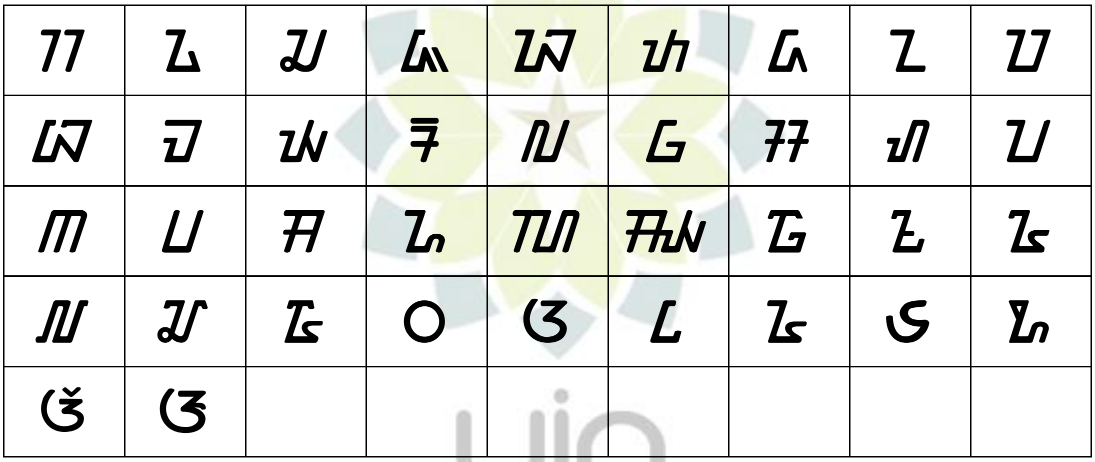
Perbandingan dilakukan untuk menggetahui tingkat akurasi dan tingkat kecepatan proses dari kedua algoritma tersebut, algoritma *feature extraction* menggunakan pengenalan berdasarkan ciri-ciri khusus yang dimiliki oleh suatu citra, sedangkan template matching mengenali suatu citra berdasarkan citra biner yaitu 0 dan 1 yang dimiliki oleh citra tersebut yang nantinya disesuaikan dengan data latih.

Pada penelitian sebelumnya dalam jurnal berjudul "Implementasi *Optical Character Recognition* pada Kamus aksara Sunda-Indonesia Menggunakan

Algoritma *Feature Extraction* Berbasis Android" terdapat beberapa hal yang harus di kembangkan diantaranya yaitu tidak dapat mengenali tulisan sunda berupa angka.

3.2 Analisis Data

Pada penelitian ini data yang akan diolah merupakan gambar aksara sunda, aksara sunda tersebut dapat berasal dari pengambilan gambar kamera ponsel maupun dari galeri ponsel, data aksara sunda tersebut seperti pada Gambar 3.1 berikut :



Gambar 3.1 Analisis Data

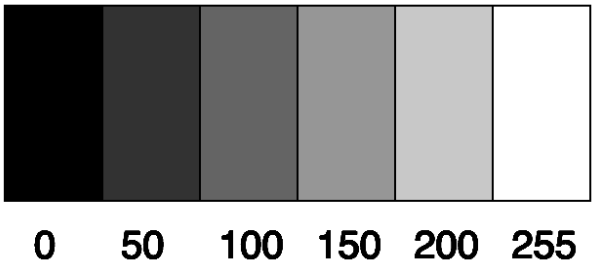
3.3 Analis Pemecahan Masalah

Berdasarkan analisis masalah pada poin 3.1, terdapat beberapa solusi untuk mengatasinya yaitu diantaranya untuk mengurangi suatu *noise* untuk data aksara sunda diatas, aksara sunda yang menjadi inputan terlebih dahulu masuk tahapan *preprocessing* yaitu tahap pengolahan gambar sebelum gambar masuk pada tahap *recognition* oleh algoritma *template matching* dan *feature extraction.* Tahap *preprocessing* ini memerlukan penambahan algoritma untuk memperbaiki citra, algoritma tersebut meliputi *luminosity* dan *tresholding* dan Untuk mengatasi

angka yang tidak terbaca maka data latih aksara sunda akan ditambah agar menambah tingkat akurasi data angka.

3.4 Analisis Perubahan R,G,B ke Citra Greyscale (Algoritma *Luminosity*)

Gambar inputan yang masih berupa gambar dengan warna RGB harus dikonversi terlebih dahulu kedalam citra *grayscale*, citra grayscale merupakan citra dengan nilai R = G = B, yang artinya tingkat intensitas dari setiap warnanya sama, memiliki intensitas 0 sampai 255 seperti pada gambar 3.2 berikut :

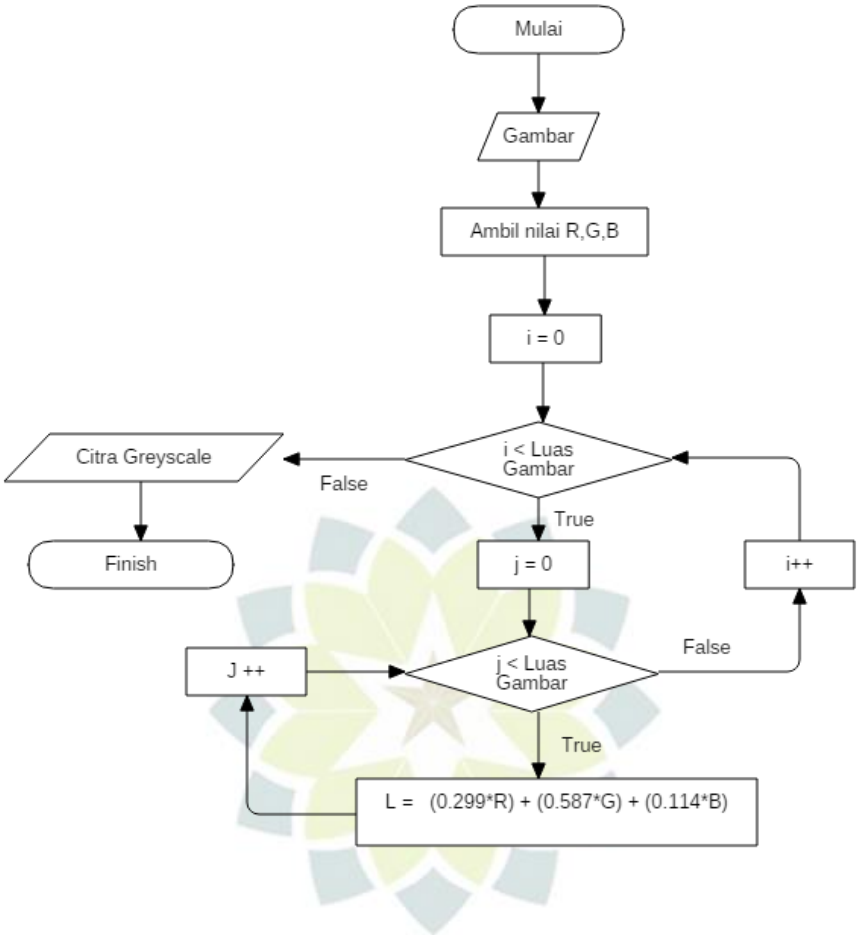


Gambar 3.2 Intensitas Citra *Greyscale*

Semakin besar nilai nya maka akan semakin mendekati warna putih, dan semakin kecil nilainya maka akan mendekati warna hitam, untuk mengubah citra warna RGB ke citra *Greyscale* dapat menggunakan algoritma *luminosity* dengan rumus persamaan berikut :

$$\textbf{CitraGreyscale} = (0.299*R) + (0.587*G) + (0.114*B)$$

Flowchart untuk proses *luminosity* yang bertujuan merubah citra RGB ke Greyscale dapat dilihat pada gambar 3.3.



Gambar 3.3 *Flowchart* Perubahan citra RGB ke *Greyscale*

Proses *Greyscale* ini merupakan tahap awal *preprocessing* sebelum masuk pada tahap binerisasi gambar. Pada gambar 3.4 dibawah ini merupakan contoh hasil perubahan gambar ke *greyscale.*

Gambar 3.4 perubahan citra RGB kr *Greyscale*

3.5 Analisis Proses Perubahan Citra *Greyscale* ke Biner (dengan *Thresholding*)

Citra Biner merupakan citra yang diproses melalui pemisahan piksel, pemisahannya dilakukan berdasarkan derajat keabuan objek tersebut. Citra biner

hanya memiliki dua warna yaitu warna hitam dan warna putih, untuk menyimpan warna ini dibutuhkan 1 bit *memory*. Pada citra biner ini setiap warna memiliki nilai yaitu warna putih bernilai 0 sedangkan warna hitam bernilai 1.

mengkonversi citra *Greyscale* ke citra Biner dapat menerapkan cara mengatur nilai *Threshold* nya (Nilai Ambang), Nilai ambang merupakan nilai citra *greyscale* bernilai 0 sampai 255, dengan menentukan nilai ambang kita dapat mengatur citra mana yang akan masuk nilai 0 (Putih) dan citra mana yang akan masuk nilai 1 (Hitam). Proses perubahan citra dapat di rumuskan dengan persamaan berikut :

$$b(i) = \begin{cases} 0, & i \geq a \\ 1, & i < a \end{cases}$$

Gambar 3.5 Persamaan Biner

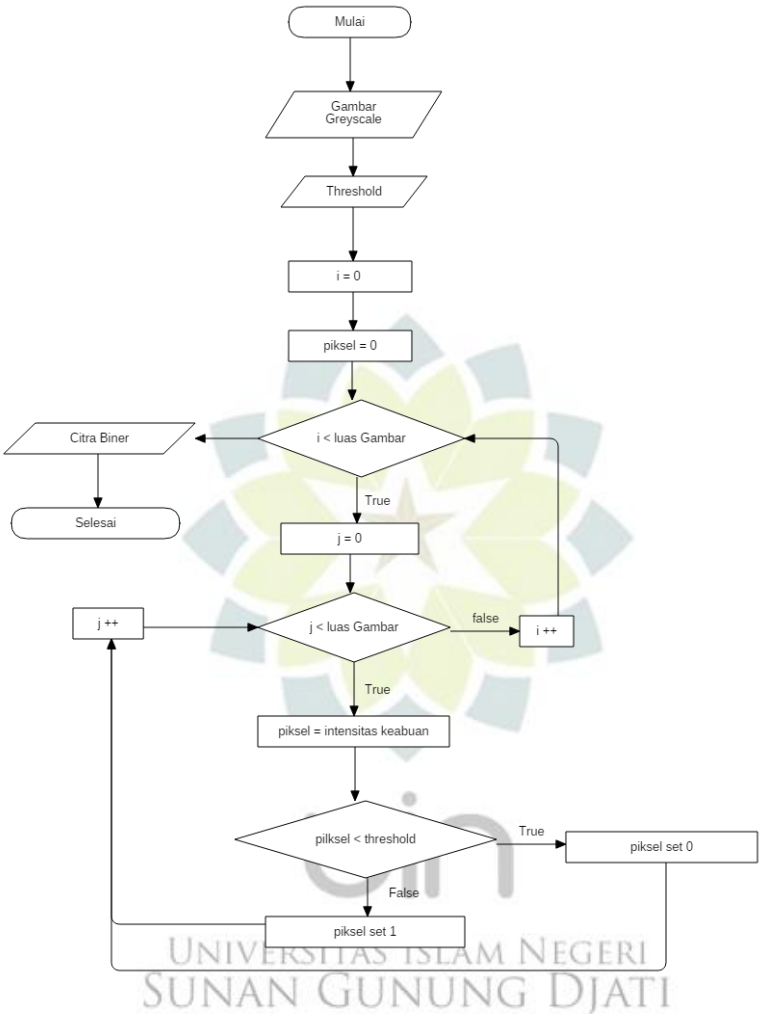
Keterangan :

a = *Thresholder* (Ambang Batas)

I = intensitas Piksel

b = biner yang dicari

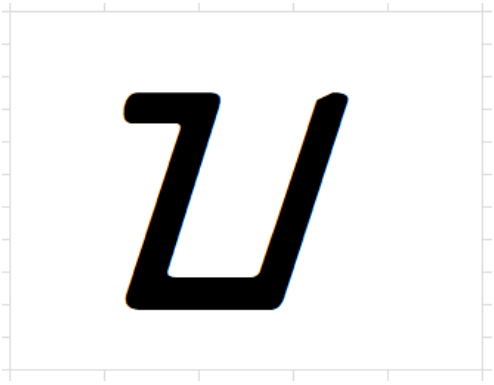
Flowchart untuk proses perubahan citra *greyscale* ke citra biner dapat dilihat pada gambar 3.6.



Gambar 3.6 *Flowchart* perubahan citra *Greyscale* ke Biner

3.6 Analisis Pattern Recognition

Proses *Pattern Recognition* merupakan proses lanjutan setelah pengolahan gambar dari R, G, B ke Greyscale dan dari Greyscale ke Biner, pada tahap ini gambar yang sudah berbentuk biner akan dikenali berdasarkan pola yang telah tersusun seperti pada gambar 3.7.



0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	1	1	0
0	0	0	1	1	1	1	0	0	0	0	1	1	0
0	0	0	0	0	1	1	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

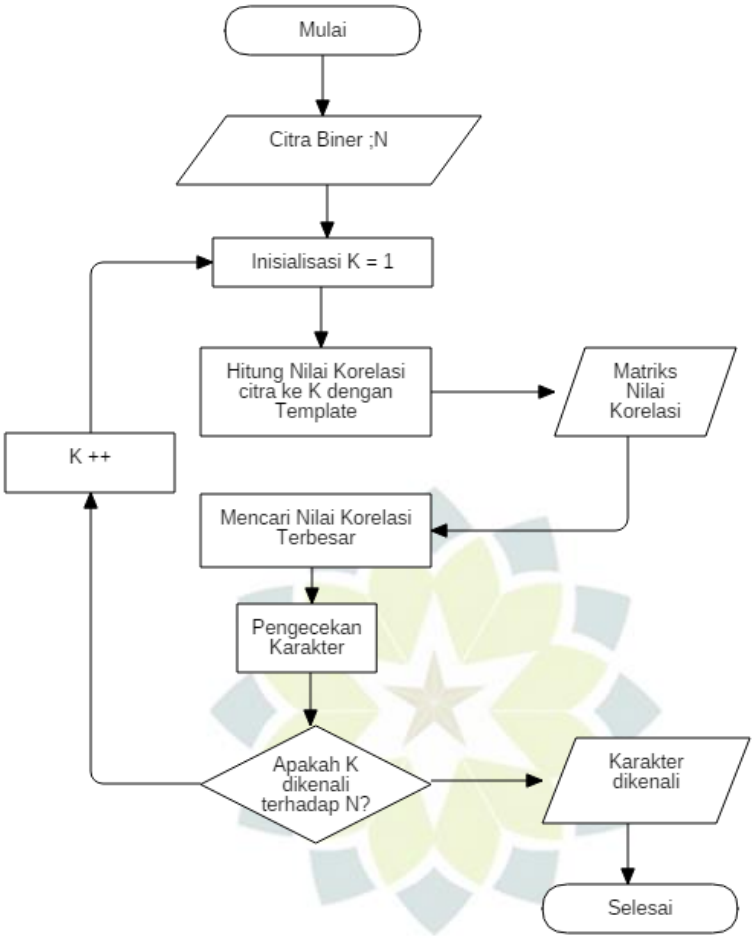
Gambar 3.7 *Pattern Recognition*

Terdapat dua algoritma untuk mengenali sebuah citra, yaitu *Feature Extraction* dengan algoritma *Template Matching*.

3.6.1 Algoritma *Template Matching*

Algoritma *template matching* merupakan algoritma yang memiliki konsep melakukan pencocokan setiap pixel dari gambar yang bercitra Biner terhadap *template* dari data latih yang telah tersedia.

Gambar 3.8 dibawah ini mendeskripsikan alur dari algoritma template matching dimana citra biner yang menjadi inputan akan dihitung nilai korelasinya (Nilai kecocokan) nilai korelasi yang terbesar yang dianggap sesuai dengan *template* [18], Pada pengenalannya *template matching* mengenali dengan cara menghitung nilai korelasi antara citra *input* dengan citra *template.*



Gambar 3.8 *Flowchart Template Matching*

Menghitung nilai korelasi bisa dengan persamaan rumus pada gambar 3.9.

$$r = \frac{\sum_{k=1}^{n}(Xik - Xi).(Xjk - Xj)}{\sqrt{\sum_{k=1}^{n}(Xik - Xi)^2 \cdot \sum_{k=1}^{n}(Xjk - Xj)^2}}$$

Dimana :

$$Xj = \frac{1}{n}\sum_{k=1}^{n} Xjk\,,$$

$$Xi = \frac{1}{n}\sum_{k=1}^{n} Xik$$

Gambar 3.9 Rumus *Template Matching Corelation* [18].

Keterangan :

n = jumlah pixel dalam suatu matriks.

r = nilai korelasi dua buah matriks.

Xij = nilai pixel indeks K matriks J (matriks template).

Xik = nilai pixel indeks K matriks I (matriks input).

Xj = nilai rata-rata pixel matriks J.

Xi = nilai rata-rata pixel matriks I.

Contoh implementasi :

Diketahui data Biner suatu inputan seperti pada gambar 3.10 Dibawah ini :

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	0	0	1	1	0	0	0	1	1	1	0
0	0	0	0	1	1	0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 3.10 Biner inputan.

untuk mencari gambar yang sesuai dengan citra input, maka dilakukan proses recognition dengan *Template Matching Correlation* terhadap *template* data latih yang tersedia, nilai korelasinya seperti berikut :

a. Data input terhadap *template* pertama

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	0	0	1	1	0	0	0	1	1	1	0
0	0	0	0	1	1	0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	1	1	1	1	0
0	0	1	1	1	1	0	0	0	1	1	1	1	0
0	0	0	0	1	1	0	0	0	0	0	1	1	0
0	0	0	1	1	0	0	0	0	0	1	1	0	0
0	0	1	1	0	0	0	0	0	1	1	0	0	0
0	1	1	1	0	0	0	0	1	1	1	0	0	0
1	1	1	0	0	0	0	1	1	1	0	0	0	0
1	1	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

a. Biner Citra *Template* 1 Biner Citra Input

Nilai Korelasi yang di dapat antara citra *template* dengan data input di atas setelah dihitung dengan rumus *Template Matching Correlation* diatas adalah sebesar 0,634012.

b. Data Input terhadap *template* kedua

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	0	0	1	1	0	0	0	0	1	1	0
0	0	0	0	1	1	0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Biner Citra *Template* 2

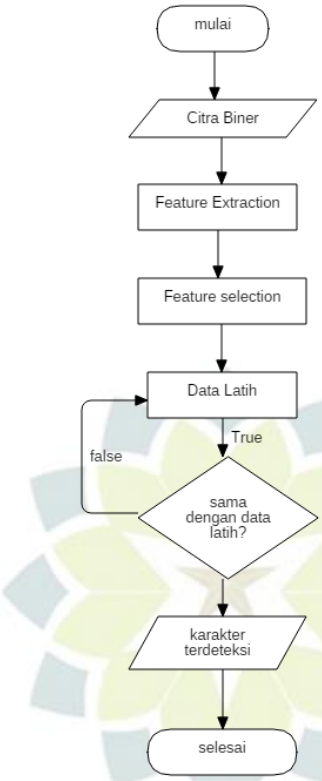
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	0	0	1	1	0	0	0	1	1	1	0
0	0	0	0	1	1	0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Biner Citra Input

Pada perhitungan citra input dengan citra *template* 2 yang telah dihitung dengan rumus template *Matching Correlation* mendapatkan nilai korelasi sebesar 0,86711.

Dari hasil pengenalan diatas antara data input dengan data *template* 1 dan data *template* 2, dimana data *template* 1 memperoleh nilai korelasi sebesar 0,634012 sedangkan data template 2 memperoleh nilai korelasi sebesar 0,86711 maka data yang paling cocok adalah data dengan nilai korelasi terbesar yaitu data *template* kedua, maka huruf inputan dikenali sebagai data *template* kedua.

3.6.2 Algoritma *Feature Extraction*



Gambar 3.11 *Flowchart* Algoritma *Feature Extraction*

Feature Extraction merupakan algoritma yang melakukan pengenalan berdasarkan ciri-ciri khusus yang dimiliki [6]. Ciri-ciri khusus yang dimiliki akan disamakan dengan data latih yang tersedia. Algoritma *feature extraction* memiliki beberapa ciri-ciri yang di klasifikasi yaitu diantaranya Perbandingan lebar dan tinggi (*Ratio*), Jumlah garis Horizontal, Jumlah garis vertikal, citra karakter terbuka (atas, bawah, kanan, kiri), perpotongan garis vertikal ditengah citra, perpotongan garis horizontal ditengah citra. Untuk klasifikasinya menggunakan

pendekatan Algoritma K-*Nearst Neighbour* (KNN), dimana hasil citra yang di dapat ini akan dibandingkan sesuai dengan tetangga terdekatnya (data *template* yang paling mendekati). Tabel 3.1 merupakan contoh pengenalan Citra template *feature Extraction*.

Tabel 3.1 *Feature Extraction* Data *Template*

no	aksara	*ratio*	*Open*				*Intersection*		*Line*	
			O.R	O.L	O.T	O.B	I.V	I.H	L.H	L.V
1		1.25	0	1	1	0	0	0	3	2
2		1.25	1	1	0	0	0	0	2	1
3		1.11	0	0	0	2	0	0	1	3

Keterangan : O.R = *Open Right.*

O.L = *Open Left.*

O.T = *Open Top.*

O.B = *Open Bottom.*

I.V = *Intersection Vertical*

I.H = I*ntersection Horizontal.*

L.H = *Line Horizontal.*

L.V = *Line Vertical*

Citra Masukan dari inputan Sebagai berikut :

no	aksara	*ratio*	*Open*				*Intersection*		*Line*	
			O.R	O.L	O.T	O.B	I.V	I.H	L.H	L.V
1	..	1.11	0	1	1	0	0	0	3	2

Tabel 3.2 *Feature Extraction Data Input*

Pengenalan dengan *K-NN* untuk klasifikasi yaitu dengan mengambil citra *feature extraction*, ciri-ciri citra masukan *feature extraction* tersebut akan dicari

yang cocok dengan ciri-ciri citra *feature extraction* data *template*, perhitungan *K-NN* Untuk klasifikasi adalah sebagai berikut :

a. Klasifikasi data masukan terhadap template **𝑼** :

$$\text{perhitungan} = \sqrt{\begin{matrix}(1.25-1.11)^2+(0-1)^2+(1-1)^2+(1-1)^2+\\(0-0)^2+(0-0)^2+(0-0)^2+(3-3)^2+(2-2)^2\end{matrix}}$$

$$= \sqrt{0.0196} = 0{,}14$$

b. Klasifikasi data masukan terhadap template **𝑳** :

$$\text{perhitungan} = \sqrt{\begin{matrix}(1.25-1.11)^2+(1-1)^2+(1-1)^2+(0-1)^2+\\(0-0)^2+(0-0)^2+(0-0)^2+(2-3)^2+(1-2)^2\end{matrix}}$$

$$= \sqrt{3{,}0196} = 1{,}7$$

c. Klasifikasi data masukan terhadap template **𝑴** :

$$\text{perhitungan} = \sqrt{\begin{matrix}(1.25-1.11)^2+(0-1)^2+(0-1)^2+(0-1)^2+\\(2-0)^2+(0-0)^2+(1-0)^2+(1-3)^2+(3-2)^2\end{matrix}}$$

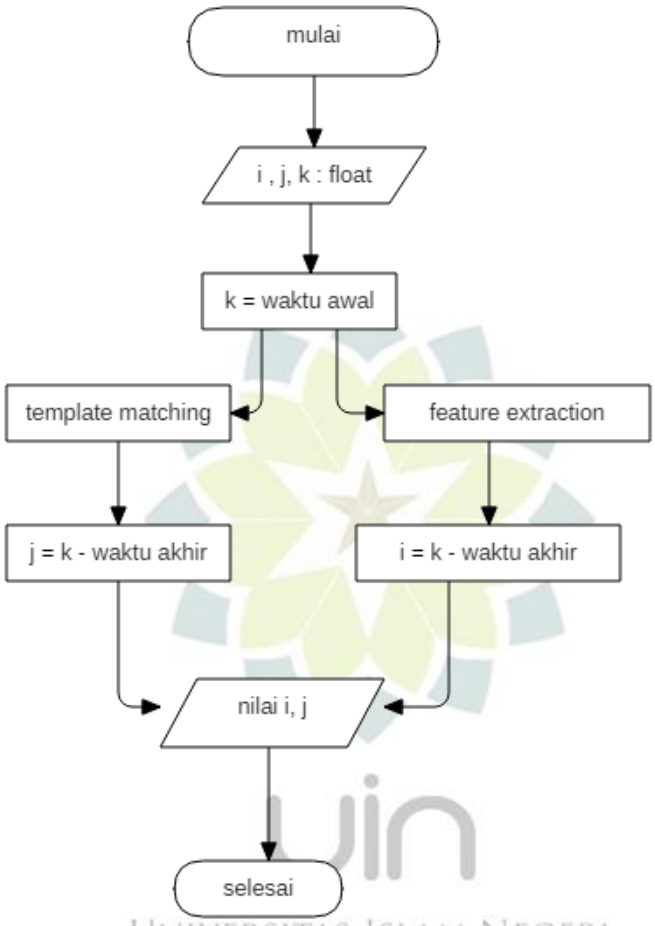
$$= \sqrt{13{,}0196} = 3{,}6$$

Dari Perhitungan K-NN diatas dapat di sortir secara Ascending maka urutannya menjadi : 0.14, 1.7 dan 3.6, maka yang paling mendekati kepada data template adalah *value* 0.14, maka citra inputan dikenali sebagai aksara **𝑼.**

3.7 Analisis Perbandingan Algoritma

Tahap ini merupakan tahap untuk membandingkan algoritma yang dipakai, yaitu algoritma *feature extraction* dengan algoritma *template matching*. Perbandingan algoritma akan meliputi perbadingan waktu proses dan perbandingan akurasi, akurasi merupakan kedekatan antara nilai benar dengan data uji [19]. Dan waktu proses algoritma yaitu waktu yang diperlukan dari awal

algoritma di jalankan sampai algoritma selesai di jalankan, Gambar 3.2 merupakan flowchart dari perbandingan algoritma**.**



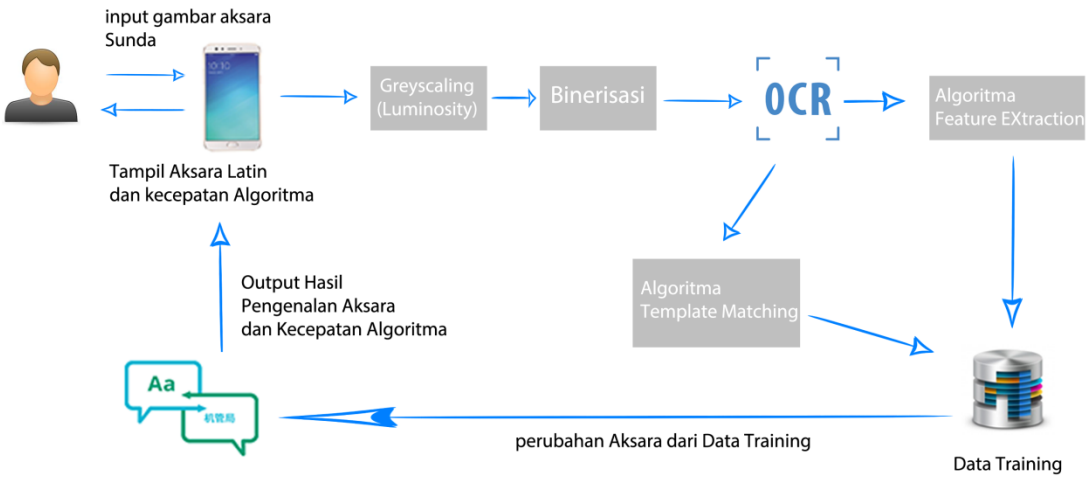
.

Gambar 3.12 *Flowchart* perbandingan Algoritma

Perhitungan tingkat akurasi dapat menggunakan rumus seperti dibawah ini.

$$\text{Akurasi} = \frac{jumlah\ karakter\ benar}{total\ karakter}\ X\ 100\%$$

3.8 Arsitektur Sistem



Gambar 3.13 Arsitektur Sistem

Pada gambar 3.13 di atas merupakan skema sistem yang akan dibangun dimana User menginput dari kamera handphone ataupun dari galeri gambar terhadap objek teks aksara sunda yang selanjutnya gambar yang telah diinput akan masuk pada proses greyscaling dengan algoritma *Luminosity*, proses *greyscaling* ini berfungsi untuk mengetahui tingkat derajat keabuannya, setelah proses *greyscaling* proses selanjutnya adalah proses binerisasi, pada proses binerisasi gambar citra abu dirubah menjadi gambar yang hanya memiliki dua intensitas warna, yaitu hitam dan putih dengan kode biner 1 untuk hitam dan 0 untuk putih, tahap selanjutnya gambar dengan citra biner dibaca oleh OCR dimana data *Image* dari OCR akan di sesuaikan dengan Algoritma *Template Matching* atau dengan algoritma *Feature Extraction* yang nantinya masuk ke data training berdasarkan pencocokan masing-masing data tersebut sesuai algoritmanya, dan ketika menemukan data yang cocok maka output akan dibaca *user* berupa aksara latin dari hasil pengenalan aksara sunda.

3.9 Perancangan Sistem

Perancangan sistem menggunakan suatu pemodelan UML (*Unified Modeling Language*) yang dimana digunakan untuk pendekatan sistem yang yang mengaplikasikan pemrograman berorientasi objek.

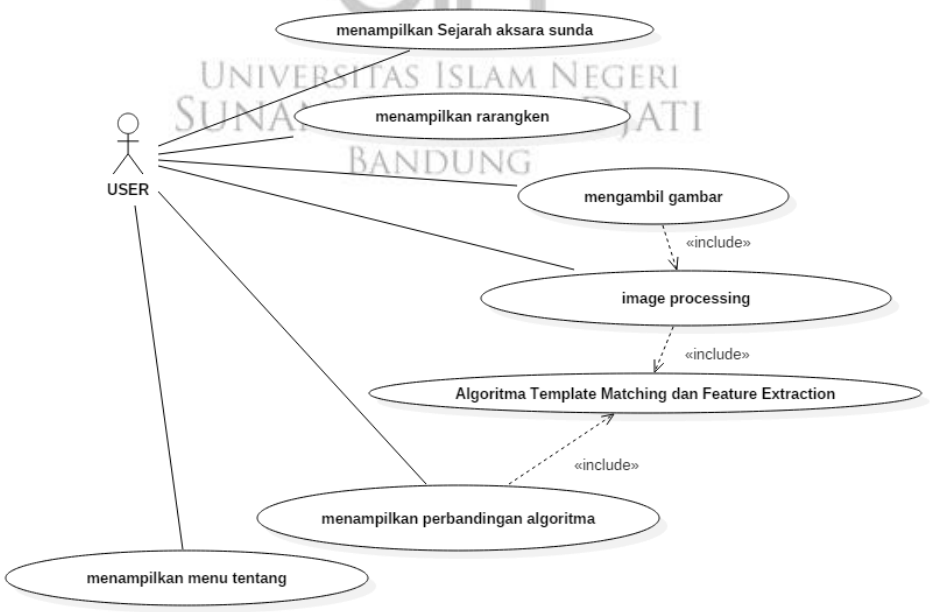
a. *Use Case diagram*

1. Definisi Aktor

Tabel 3.3 Definisi Aktor

No	Actor	Deskripsi
1.	*User*	*User* pada rule ini bisa menggunakan semua fitur yang tersedia dengan mengidentifikasi data sample dan melakukan *testing*.

2. Use Case Diagram



Gambar 3.14 *Use Case Diagram*

Gambar 3.14 diatas menggambarkan *use case* diagram yang dibangun oleh sistem, Pada use case diagram menjelaskan bagaimana suatu sistem berinteraksi, dan apa saja yang bisa dilakukan oleh sistem tersebut.

3. Skenario *Use Case*

Skenario *Use Case* merupakan gambaran dari alur sistem dimana setiap interaksi di gambarkan dari sudut pandang aktor. Berikut merupakan skenario *use case diagram.*

Tabel 3.4 Skenario menampilkan sejarah aksara Sunda

No.	Case - 01
Nama *Use Case*	Menampilkan sejarah aksara sunda
Aktor	*User*
Tujuan	Untuk menapilkan sejarah Aksara Sunda
Deskripsi	1. Buka aplikasi 2. Pilih menu sejarah aksara sunda 3. Menampilkan sejarah aksara sunda

Tabel 3.5 Skenario menampilkan Rarangken

No.	Case - 02
Nama *Use Case*	Menampilkan Rarangken
Aktor	*User*

Tujuan	Untuk menampilkan macam-macam rarangken pada aksara sunda.
Deskripsi	1. Buka aplikasi 2. Pilih menu rarangken sunda 3. Menampilkan tiga belas rarangken dalam aksara sunda.

Tabel 3.6 Skenario Mengambil gambar

No.	Case - 03
Nama *Use Case*	Mengambil gambar
Aktor	*User*
Tujuan	Untuk mengambil gambar dari galeri ataupun dari kamera *handphone*.
Deskripsi	1. Buka aplikasi. 2. Pilih menu OCR Sunda – Latin. 3. Pilih upload Foto. 4. Foto yang diambil disimpan dalam perangkat android.

Tabel 3.7 Skenario *Image Processing*

No.	Case - 04
Nama *Use Case*	*Image Prosessing*
Aktor	*User*
Tujuan	Untuk mengubah citra gambar yang telah diinput menjadi citra Biner
Deskripsi	1. Buka aplikasi. 2. Pilih menu OCR Sunda – Latin. 3. Pilih upload Foto. 4. Gambar yang diambil disimpan dalam perangkat android. 5. Lakukan pemotongan pada gambar. 6. Klik selanjutnya. 7. Menampilkan citra biner dari gambar yang telah di potong

Tabel 3.8 Skenario menampilkan Perbandingan Algoritma

No.	Case - 05
Nama *Use Case*	Menampilkan perbandingan Algoritma
Aktor	*User*
Tujuan	Untuk mengetahui perbandingan Algoritma antara

	template matching dengan *feature extraction*
Deskripsi	1. Buka aplikasi. 2. Pilih menu OCR Sunda – Latin. 3. Pilih upload Foto. 4. Gambar yang diambil disimpan dalam perangkat android. 5. Lakukan pemotongan pada gambar. 6. Klik selanjutnya. 7. Menampilkan citra biner dari gambar yang telah di potong. 8. Klik selanjutnya. 9. Menampilkan Tulisan sunda dalam aksara latin dan proses dari waktu algoritma.

Tabel 3.9 Skenario menampilkan Menu Tentang

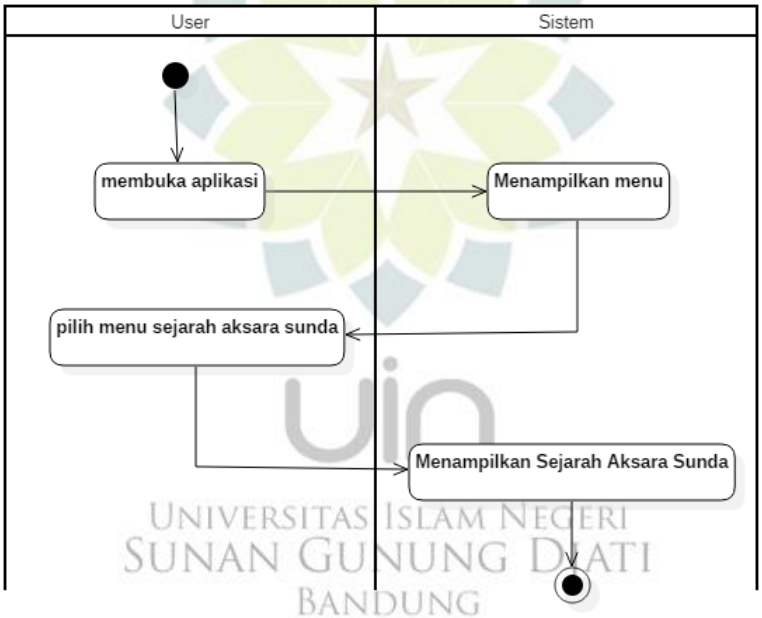
No.	Case - 06
Nama *Use Case*	Menampilkan Menu Tentang
Aktor	*User*
Tujuan	Untuk mengetahui tentang aplikasi yang dibuat.
Deskripsi	1. Buka aplikasi. 2. Pilih menu tentang. 3. Menampilkan tentang informasi aplikasi.

b. *Activity Diagram*

activity diagram merupakan gambaran dari aktivitas yang sedang berlangsung pada sistem yang dibuat. *activity* diagram dapat berupa *decision* yang kemungkinan bisa terjadi ataupun berbentuk *paralel.* Berikut *activity* diagram dari sistem yang dibuat.

1. *Activity diagram* menampilkan sejarah sunda.

Pada gambar 3.15 berikut merupakan activity diagram dari menampilkan aksara sunda.

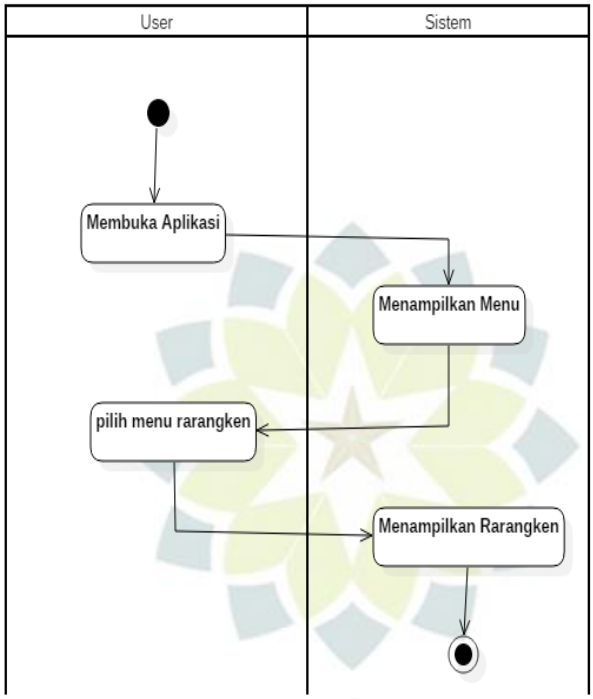


Gambar 3.15 *Activity Diagram* menampilkan aksara sejarah aksara sunda

Pada gambar 3.15 *activity diagram* diatas, merupakan tahap untuk menampilkan sejarah aksara sunda, user membuka aplikasi terlebih dahulu kemudian sistem akan menampilkan menu utama, lalu user mnekan sejarah aksara sunda selanjutnya sistem akan menampilkan sejarah aksara sunda.

2. Activity diagram menampilkan rarangken.

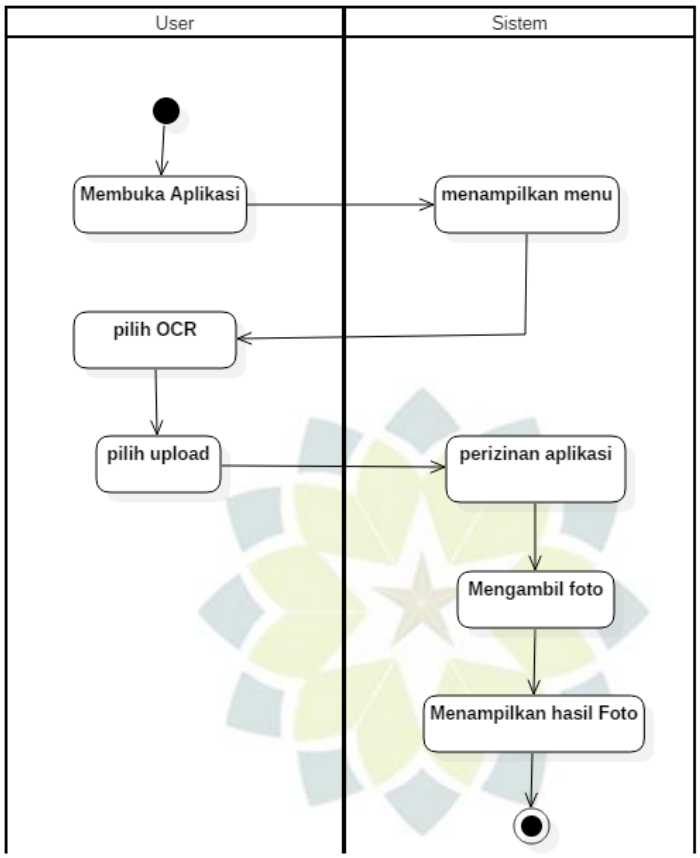
Pada gambar 3.16 berikut merupakan *activity diagram* dari menampilkan menu rarangken.



Gambar 3.16 *Activity Diagram* menampilkan Rarangken

Gambar 3.16 diatas merupakan *activity* diagram dari proses menampilkan menu rarangken, *user* pertama kali membuka aplikasi lalu sistem akan menampilkan menu utama *user* memlilih menu rarangken dan selanjutnya sistem akan menampilkan menu rarangken.

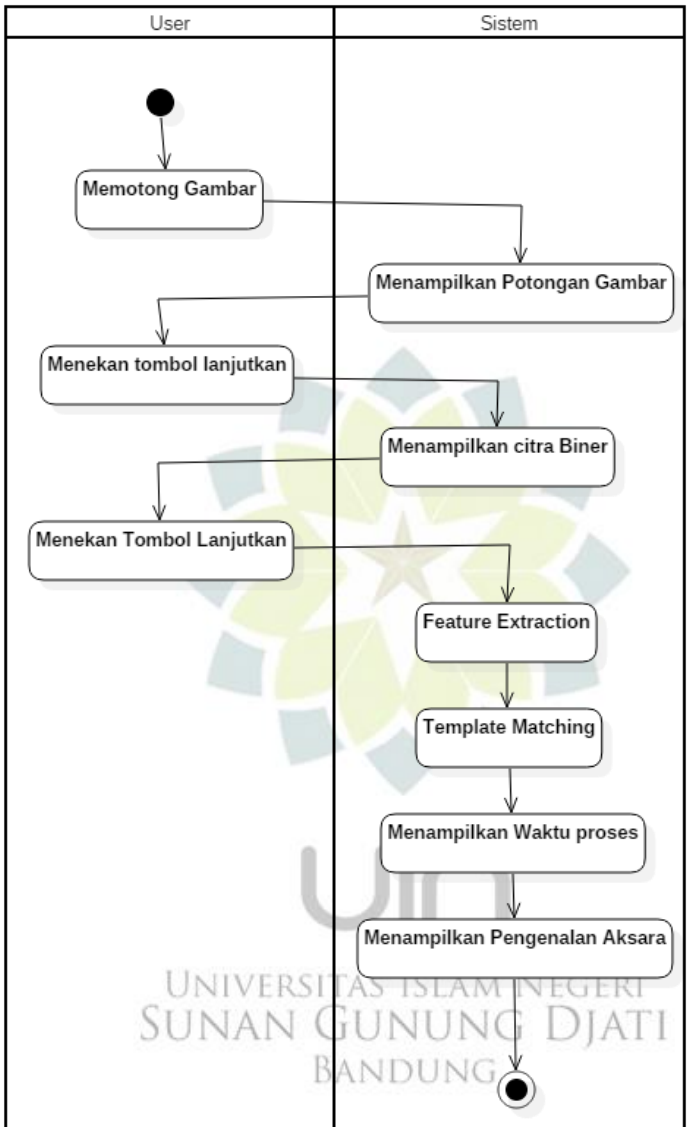
3. *Activity* diagram mengambil foto.



Gambar 3.17 *Activity Diagram* mengambil foto

Pada gambar 3.17 diatas merupakan proses pengambilan foto, yang dimana proses nya user memilih menu OCR pada pilihan menu utama, lalu user menekan tombol upload, setelah itu maka sistem akan melakukan pengecekan *permission* terhadap *handphone*, setelah itu maka akan masuk ke aktivitas pengambilan gambar atau foto dari galeri, lalu gambar yang sudah dipilih akan ditampilkan oleh sistem.

4. Activity diagram *image Processing*

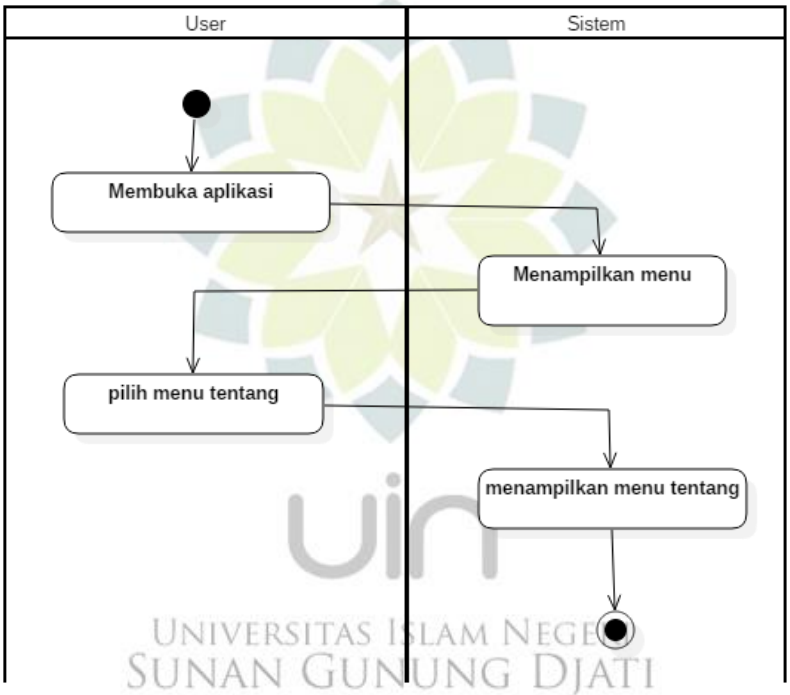


Gambar 3.18 *Activity Diagram* pengenalan aksara sunda

Pada gambar 3.18 diatas merupakan *activity* dari pengenalan aksara sunda, pada proses ini dimulai dari gambar yang sudah dipilih dilakukan pemotongan terhadap objek yang ingin dipindai lalu sistem akan menampilkan gambar yang sudah dipotong, pada aktivitas selanjutnya gambar akan di proses menjadi citra

Biner guna mempermudah algoritma dalam pembacaannya. Setelah menjadi citra biner user meneekan tombol lanjutkan dan akan masuk pada proses pengenalan oleh sistem melalui algoritma *feature extraction* dan algoritma *template matching*. Lalu selanjutnya sistem akan menampilkan hasil pengenalan berupa aksara latin, dan juga menampilkan lama waktu proses yang di peroleh.

5. *Activity* diagram tentang aplikasi

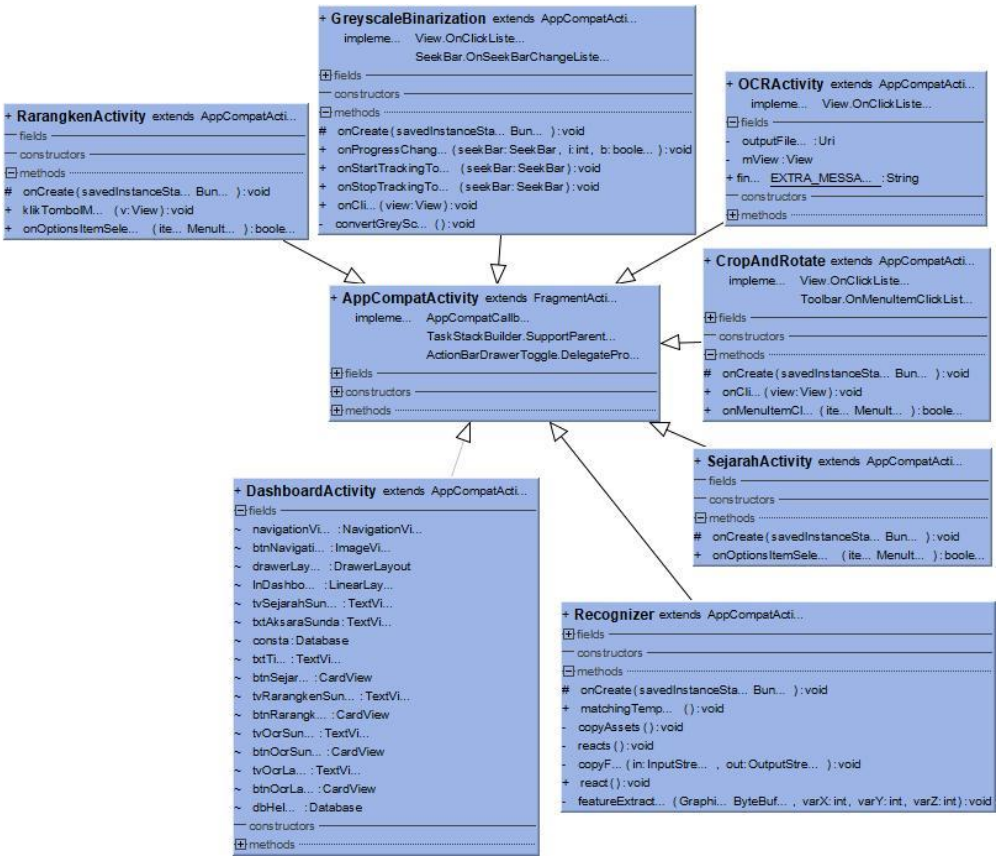


Gambar 3.19 Activity Diagram Tentang aplikasi

Pada gambar 3.19 diatas, activity menu tentang aplikasi dimana user membuka aplikasinya terlebih dahulu, lalu sistem akan menampilkan menu utama, setelah itu user memilih menu tentang dan selanjutnya sistem akan menampilkan menu tentang.

c. *class diagram*

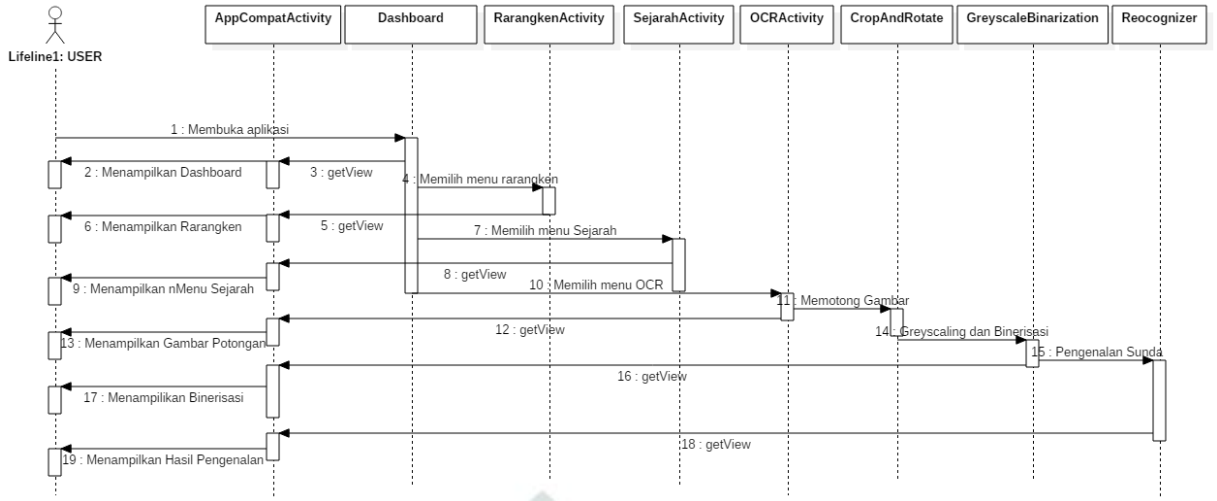
class diagram merupakan pemodelan dari setiap kelas yang terdapat pada sistem yang dirancang, didalam kelas diagram terdapat hubungan suatu kelas dengan kelas yang lainnya,gambar 3.20 berikut merupakan kelas diagram dari sistem yang sedang dirancang.



Gambar 3.20 *Class Diagram* aplikasi

d. *Sequence Diagram*

sequence diagram merupakan gambaran interaksi yang terjadi pada sebuah *object* yang dimana menunjukan rangkaian pesan yang dikirim antara *object* dalam menghasilkan suatu *output* dalam sistem. Berikut *sequence* diagram dari sistem yang dirancang.



Gambar 3.21 Sequence Diagram Aplikasi

3.10 Perancangan User Interface

Perancangan *user interface* merupakan perancangan yang berhubungan dengan tampilan yang diimplementasikan terhadap *front end* aplikasi yang berhubungan dengan pengguna. *User interface* ini sangat penting karena menjadi perantara antara manausi dengan perangkat lunak, perancangan antarmuka harus dibuat secara tepat dan baik guna untuk menjaga keutuhan informasi yang disajikan dan guna menjadikan *user* nyaman dalam pemakaian aplikasinya. [20]

1. Desain Menu *Home*

Gambar 3.22 *Desain Menu Home*

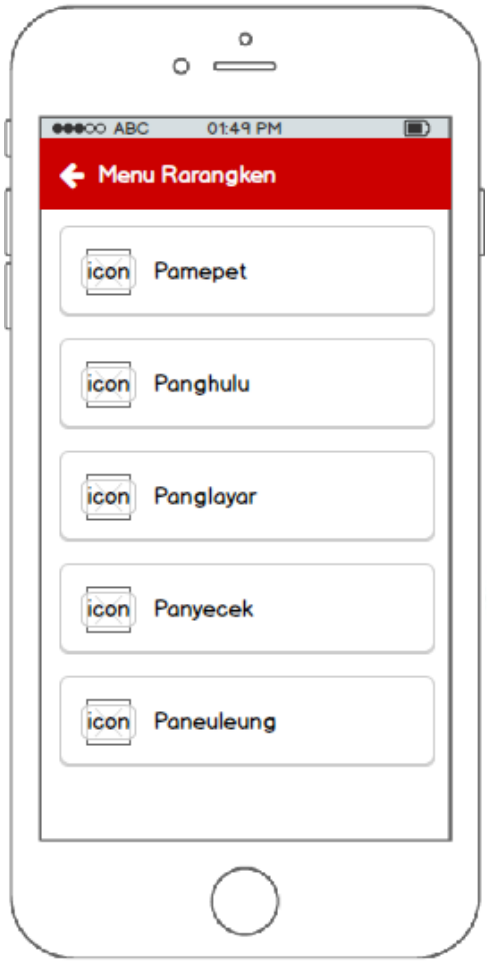
Menu *home* merupakan menu utama dimana terdapat beberapa pilihan menu lainnya, memiliki nama aplikasi "Nyunda Kuy" sebagai penarik minat *user* untuk belajar aksara sunda, di dalam menu home ini terdapat beberapa *button* dari setiap fitur yang di tawarkan. Yaitu diantaranya menu sejarah aksara sunda, menu rarangken, menu OCR (Sunda - Latin) dan menu OCR (Latin - Sunda).

2. Desain menu Sejarah aksara sunda.

Gambar 3.23 Desain menu sejarah aksara sunda

Pada gambar 3.23 diatas merupakan desain dari sejarah aksara sunda, pada desain menu ini bertujuan untuk memberikan *layouting* yang menarik untuk konten berisi aksara sunda.

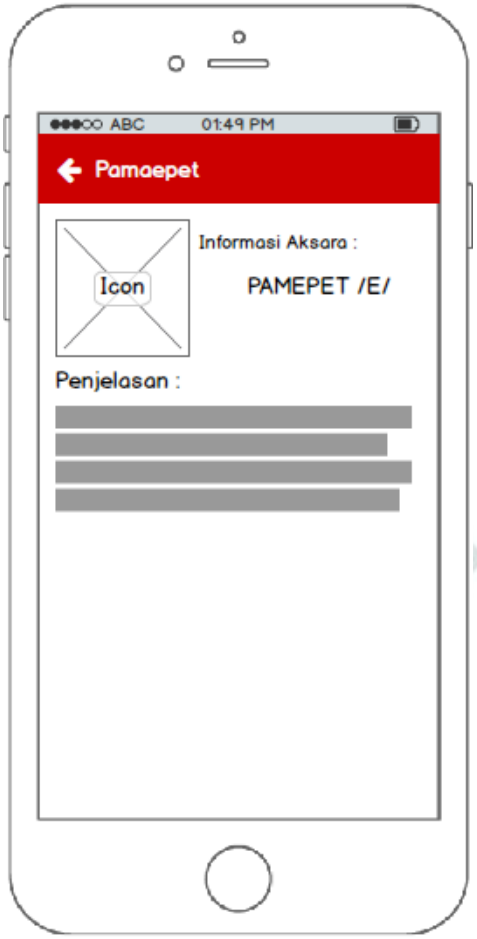
3. Desain menu *list* Rarangken



Gambar 3.24 Desain menu list rarangken

Gambar 3.24 diatas merupakan desain untuk menampilkan rarangken aksara sunda, dimana terdapat 13 rarangken yang ditampilkan pada menu ini, data rarangken ditampilkan dalam bentuk *list.*

4. Desain menu detail rarangken



Gambar 3.25 Desain menu detail

Pada menu detail rarangken diatas berfungsi untuk menampilkan materi rarangken yang telah dipilih pada menu *list* rarangken, pada menu ini menampilkan gambar rarangken aksara sunda, serta dibawahnya terdapat penjelasan dari pemakaian rarangken tersebut.

5. Desain menu *upload* gambar

Gambar 3.26 Desain Upload Gambar

Pada gambar 3.26 diatas merupakan menu untuk fitur *upload* gambar, tampilan ini muncul ketika *user* memilih menu OCR sunda-latin pada menu utama. Pada menu ini terdapat *button* berbentuk gambar yang berguna untuk aksi apabila ditekan akan membuka galeri atau kamera *handphone,* pada menu ini juga terdapat teks untuk menginformasikan bahwa yang di upload haruslah gambar aksara sunda dengan format .PNG atau .JPG.

6. Desain tampilan proses binerisasi

Gambar 3.27 Desain tampilan *Greyscaling* dan Binerisasi

Pada gambar 3.27 merupakan tampilan dari proses binierisasi, terdapat tampilan gambar yang telah di konversi ke citra biner yang sebelumnya melewati tahap *Greyscaling* (tahap konversi RGB menjadi intensitas Citra *Greyscale*), terdapat juga *seekbar* untuk mengatur nilai ambang (*Threshold*) dari suatu gambar, ketika threshold pada gambar di kurangi ataupun di tambah maka gambar citra biner akan berubah mengikuti nilai *threshold* tersebut.

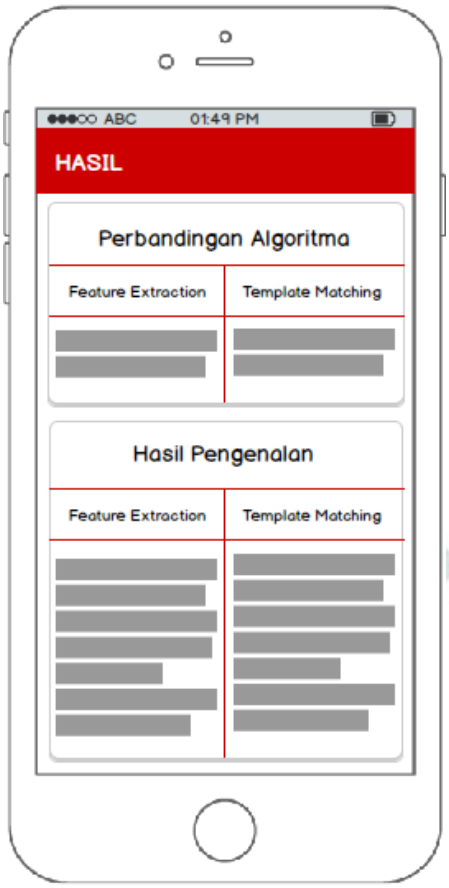
Berikut *Pseudo code* perubahan citra RGB ke *Greyscale* :

```
Algoritma Konversi_greyscale;
Deklarasi
     arrayBitmap : array [ Bitmap.size] of double
    x,y          :  integer
    r,g,b        :  integer
Begin
     FOR ( x = 1; x ← arrayBitmap.width; x++)
       FOR ( y = 1; y ← arrayBitmap.heigh; y++)

          r  ←  arrayBitmap[x][y].getR;
          g ←   arrayBitmap[x][y].getG;
          b ←   arrayBitmap[x][y].getB;
          arrayBitmap[x][y] = (0.299*r) + (0.587*g) + (0.114*b);

      END FOR
    END FOR
END
```

Berikut *Pseudo code* perubahan citra *Greyscale* ke Biner :

```
Algoritma Konversi_biner;
Deklarasi
     arrayBitmap : array [ Bitmap.size] of double
    x,y          :  integer
   threshold : integer
Begin
     read (threshold)
     FOR ( x = 1; x ← arrayBitmap.width; x++)
       FOR ( y = 1; y ← arrayBitmap.heigh; y++)
          IF arrayBitmap[x][y] < threhold then
              arrayBitmap[x][y] = 0;
          ELSE
              arrayBitmap[x][y] = 1;
         END ELSE
         END IF
      END FOR
    END FOR
END
```

7. Desain tampilan hasil



Gambar 3.28 Desain tampilan Hasil

Gambar 3.28 diatas merupakan tampilan dari hasil pengenalan aksara sunda, pada tampilan tersebut dibagi menjadi dua *layout* yaitu layout untuk perbandingan algoritma dan *layout* untuk hasil pengenalan. Algoritma yang dibandingkan yaitu algoritma *template matching* dengan algoritma *feature extraction.* Terdapat juga layout hasil dari kecepatan proses dari algoritma tersebut.

8. Desain tampilan menu Tentang Aplikasi

Gambar 3.29 Desain tampilan Tentang Aplikasi

Gambar 3.29 diatas merupakan desain dari tampilan tentang aplikasi, dimana pada tampilan tersebut memiliki informasi mengenai pembuat atau peneliti.