

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Implementasi merupakan tahap penerapan dari sistem yang telah di analisis pada bab sebelumnya, model pengembangan yang dipakai dengan model *prototype* yaitu *customers drivers mockup*.

4.1.1 Perangkat Pendukung

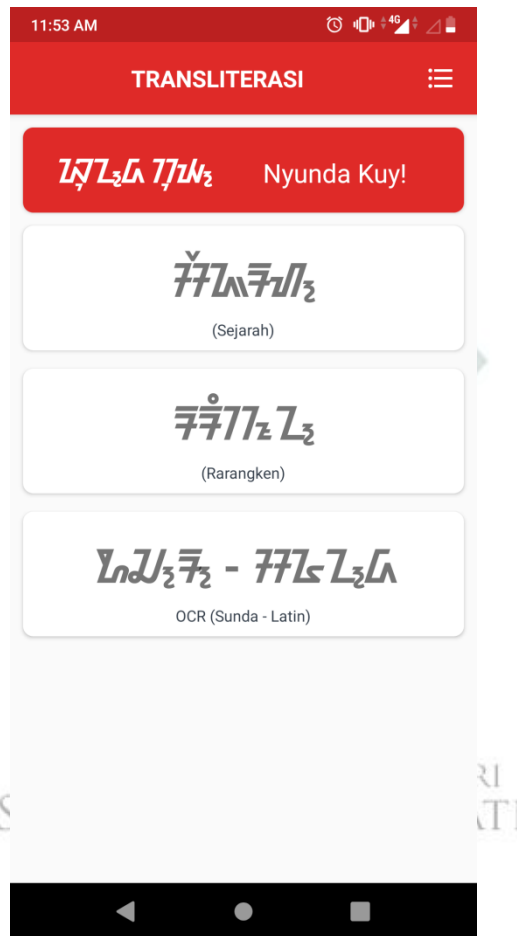
Perangkat pendukung merupakan suatu tools yang dipakai untuk membuat aplikasi tugas akhir ini, perangkat pendukung tersebut meliputi perangkat keras dan juga perangkat lunak.

1. Perangkat Keras pendukung (*Hardware*).
 - a. *Processor* Intel(R) Core(TM) i3-7130U CPU @2.7 GHZ.
 - b. RAM 8 GB
 - c. VGA 4GB.
 - d. Harddisk 1 TB.
 - e. SSD 240 GB.
 - f. *Keyboard*, monitor, dan *mouse*.
2. Perangkat Lunak Pendukung (*Software*).
 - a. Sistem Operasi Windows 10 *Home single language* 64 Bit.
 - b. Android versi 9 (*Pie*).
 - c. *Balsamiq Mockup*.
 - d. Android Studio (IDE) versi 3.4.1.
 - e. StarUML versi 5.

4.1.2 Implementasi *Interface*

Implementasi *Interface* merupakan tahap dalam penerapan *Interface* yang sudah di rancang sebelumnya.

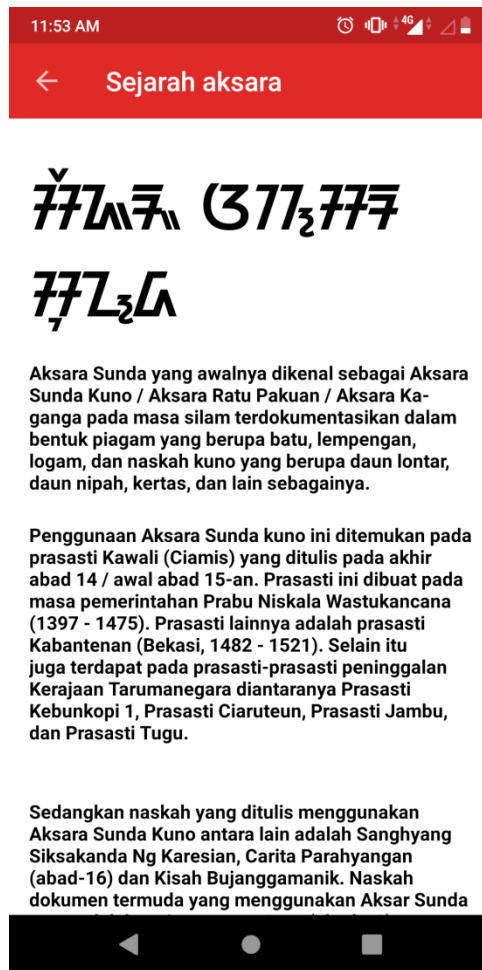
1. Tampilan menu *Home*.



Gambar 4.1 Implementasi tampilan *Home*

Pada gambar 4.1 diatas merupakan menu *home* dari aplikasi yang muncul setelah *Splash Screen*, Terdapat tiga menu yang bisa diakses yaitu diantaranya menu Sejarah, Rarangken, dan juga menu OCR(Sunda-Latin) dan juga terdapat navigation drawer disebelah kanan untuk melihat informasi tentang aplikasi.

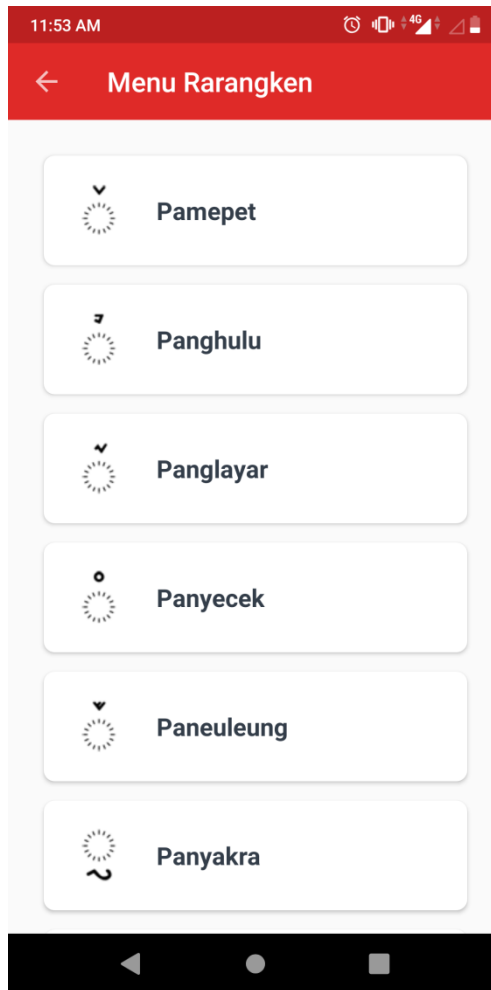
2. Tampilan menu Aksara sunda.



Gambar 4.2 Implementasi tampilan Aksara Sunda

Gambar 4.2 diatas merupakan tampilan sejarah aksara sunda, pada desain menu ini bertujuan untuk memberikan *layouting* yang menarik untuk konten berisi aksara sunda.

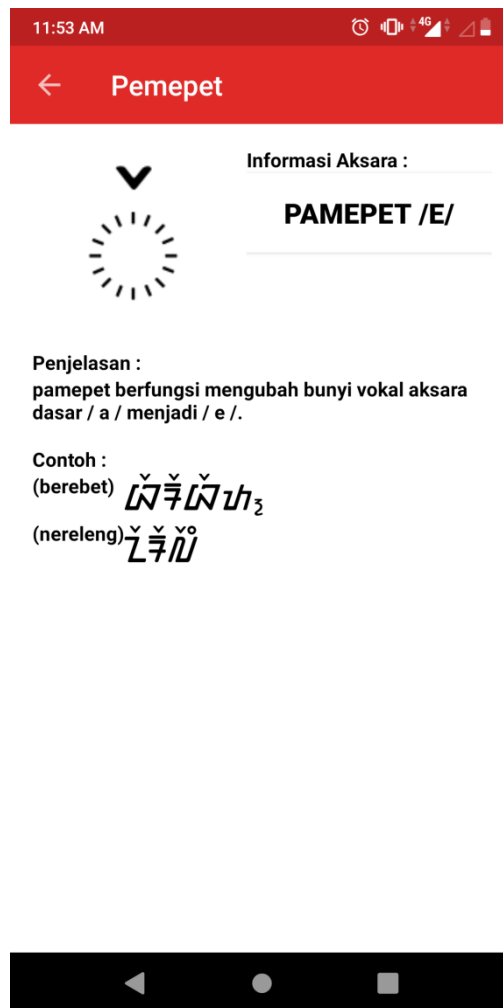
3. Menu Rarangken



Gambar 4.3 Implementasi tampilan Rarangken

Pada gambar 4.3 diatas merupakan tampilan dari menu rarangken, terdapat tiga belas menu rarangken yang ditampilkan. Rarangken ditampilkan dalam bentuk list dan dapat di klik untuk selanjutnya mengarah ke menu penjelasan rarangken.

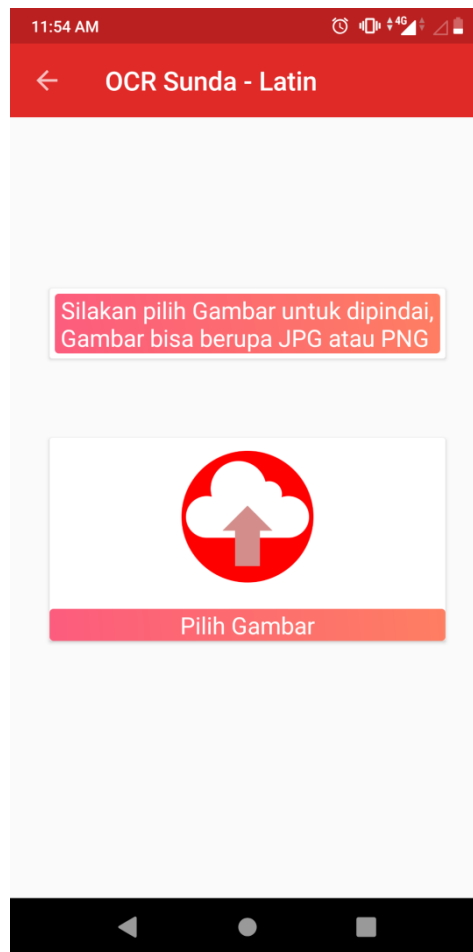
4. Tampilan penjelasan Rarangken



Gambar 4.4 Implementasi tampilan Penjelasan Rarangken

Pada gambar 4.4 diatas yaitu untuk menu penjelasan rarangken, menu ini tampil setelah *user* memilih salah satu rarangken pada menu rarangken. Pada menu ini terdapat penjelasan dari rarangken beserta contohnya.

5. Tampilan ambil Gambar



Gambar 4.5 Implementasi tampilan Ambil Gambar

Pada gambar 4.5 diatas merupakan tampilan dari menu untuk pengambilan gambar untuk nantinya diproses sebagai gambar masukan, terdapat petunjuk untuk menginputkan gambar aksara sunda dalam bentuk format png atau jpg agar prosesnya valid.

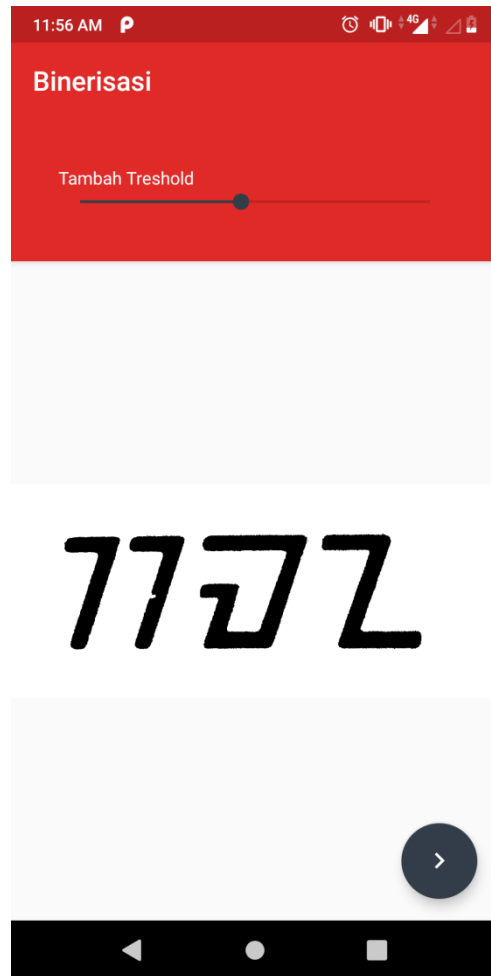
6. Tampilan *Capture* gambar



Gambar 4.6 Implementasi *Capture* Gambar

Gambar 4.6 diatas merupakan implementasi dari pengambilan *Capture* gambar, pada tampilan ini user dapat memilih aksara sunda yang akan di terjemahkan keaksara latin dapat mengeleminasi gambar yang bukan aksara sunda dengan melakukan pemotongan terlebih dahulu, terdapat pengaturan untuk rotasi gambar yang rotasinya memiliki nilai 90 derajat.

7. Tampilan *Greyscaling* dan Binerisasi Gambar



Gambar 4.7 Implementasi *Capture* Gambar

Tampilan pada gambar 4.7 diatas merupakan tampilan dari proses Binerisasi, dimana terdapat beberapa proses yaitu proses konversi citra R,G,B ke citra *Greyscale*, dan konversi citra *greyscale* ke citra Biner. Terdapat *seekbar* untuk mengatur *threshold* yang berguna untuk batasan nilai ambang citra biner dalam pemilihan citra 0 atau 1.

Source code proses Luminosity (Konversi R,G,B ke greyscale) :

```
private ByteBuffer convertGreyScale() {
    int var1 = this.mBitmap.getWidth();
    int var2 = this.mBitmap.getHeight();
    int[] var3 = new int[var1 * var2];
    this.mBitmap.getPixels(var3, 0, var1, 0, 0, var1, var2);
    byte[] var4 = new byte[var1 * var2];
    for (int var5 = 0; var5 < var3.length; ++var5) {
        var4[var5] = (byte) ((int) ((float) Color.red(var3[var5])
* 0.299F + (float) Color.green(var3[var5]) * 0.587F + (float)
Color.blue(var3[var5]) * 0.114F));
    }
    return ByteBuffer.wrap(var4);
}
```

Source code Proses Binerisasi (Konversi greyscale ke Biner) :

```
@Override
public void onStopTrackingTouch(SeekBar seekBar) {

    umbralization =
com.googlecode.leptonica.android.WriteFile.writeBitmap(
        GrayQuant.pixThresholdToBinary(pix,
Integer.valueOf(((254 * seekBar.getProgress()) / 50))
    );

    Toast.makeText(this, "Trash holdnya : " +
Integer.valueOf(((254 * seekBar.getProgress()) / 50)),
Toast.LENGTH_SHORT).show();
    img.setImageBitmap(umbralization);
}
```

8. Tampilan Perbandingan Algoritma

Perbandingan Algoritma	
Feature Extraction	Template Matching
Waktu : 65 MS	Waktu : 236 MS

Hasil Pengenalan	
Feature Extraction	Template Matching
kamana	kamana

Gambar 4.8 Implementasi Tampilan Perbandingan Algoritma

Pada gambar 4.8 diatas merupakan tampilan dari hasil perbandingan algoritma, yaitu algoritma template matching dengan algoritma feature extraction, tampilan perbandingan pertama yaitu tampilan perbandingan waktu, dan tampilan perbandingan kedua yaitu tampilan perbandingan dari hasil pengenalan aksara sunda oleh *feature extraction* dan *template matching*.

Source Code Feature Extraction :

```

private void featureExtraction(ByteBuffer Graphics, int varX, int
varY, int varZ) {
    byte[] var5;
    if (Graphics != null) {
        if (Graphics.hasArray() && Graphics.arrayOffset() == 0) {
            var5 = Graphics.array();
        } else {
            var5 = new byte[Graphics.capacity()];
            Graphics.get(var5);
        }

        ByteArrayOutputStream var6 = new ByteArrayOutputStream();
        YuvImage var7 = new YuvImage(var5, varX, varY, varZ,
(int[]) null);
        var7.compressToJpeg(new Rect(0, 0, varY, varZ), 100,
var6);
        byte[] var8 = var6.toByteArray();
        BitmapFactory.decodeByteArray(var8, 0, var8.length);
    }
    reacts();
}

```

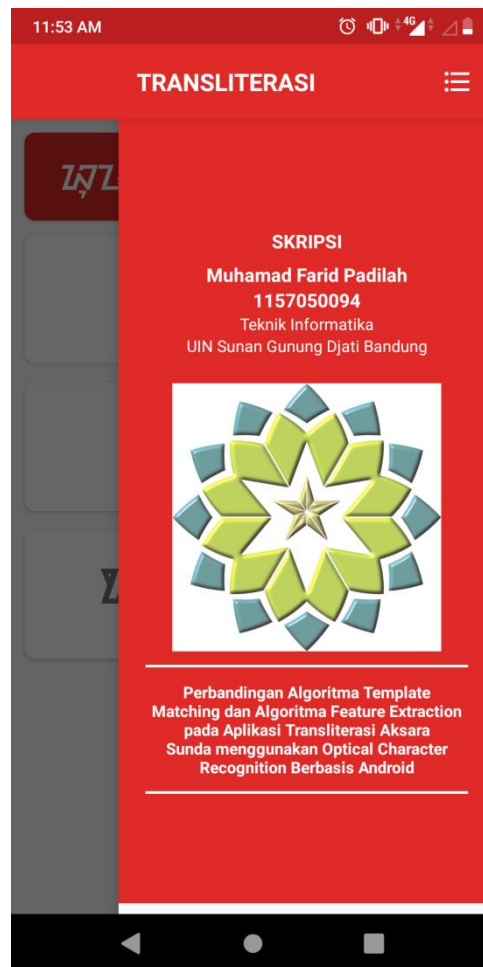
Source code Template Matching :

```

public void matchingTemplate() {
    if (!imgPath.equalsIgnoreCase("")) {
        Mat img = Imgcodecs.imread(imgPath);
        Mat temptrain = Imgcodecs.imread(templatePath);
        int result_cols = img.cols() - temptrain.cols() + 1;
        int result_rows = img.rows() - temptrain.rows() + 1;
        Mat result = new Mat(result_rows, result_cols,
CvType.CV_8UC3);
        Imgproc.matchTemplate(img, temptrain, result, matchType);
        int type = Imgproc.THRESH_TOZERO;
        Imgproc.threshold(result, result, 0.8, 1., type);
        Core.normalize(result, result, 0, 1, Core.NORM_MINMAX, -1,
new Mat());
        Core.MinMaxLocResult mmr = Core.minMaxLoc(result);
        Point matchLoc;
        if (matchType == Imgproc.TM_SQDIFF || matchType ==
Imgproc.TM_SQDIFF_NORMED) {
            matchLoc = mmr.minLoc;
        } else {
            matchLoc = mmr.maxLoc;
        }
        Imgproc.rectangle(img, matchLoc, new Point(matchLoc.x +
temptrain.cols(),
matchLoc.y + temptrain.rows()), new Scalar(0, 255,
0));
        Imgcodecs.imwrite(resPath, img);
        Mat image = Imgcodecs.imread(resPath);
        Bitmap bm = Bitmap.createBitmap(image.cols(), image.rows(),
null);
        Utils.matToBitmap(image, bm);
    }
    react();
}

```

9. Tampilan menu Tentang Aplikasi



Gambar 4.9 Implementasi Tampilan Tentang Aplikasi

Tampilan ini merupakan informasi tentang pembuat aplikasi sekaligus sebagai peneliti.

4.1.3 Pengujian Sistem

Pengujian dapat diartikan sebagai sebuah proses untuk mencari error dalam suatu sistem yang telah dibuat. Pada pengujian kali ini menggunakan model pengujian *Blackbox Testing* yaitu pengujian yang berfokus pada fungsionalitas dari aplikasi.

1. Pengujian *Interface*

a. Pengujian *Interface* Menu *Home*

Tabel 4.1 dibawah ini merupakan pengujian *blackbox* testing pada menu *home*.

Tabel 4.1 Pengujian *interface* menu *home*

No.	Skenario Pengujian	Hasil yang diharapkan	Hasil
1.	Klik tombol aksara sunda	Berpindah ke bagian detail aksara sunda	Berhasil
2.	Klik tombol rarangan	Berpindah ke bagian detail aksara sunda	Berhasil
3.	Klik tombol ocr Sunda-latin	Berpindah ke menu OCR latin Sunda	Berhasil
4.	Klik <i>Navigation Drawer</i> Tentang	Menampilkan Tentang Aplikasi	Berhasil

b. Pengujian *Interface* Menu Rarangan

Tabel 4.2 dibawah ini merupakan pengujian *blackbox* testing pada menu Rarangan.

Tabel 4.2 Pengujian *Interface* Rarangan

No.	Skenario Pengujian	Hasil yang diharapkan	Hasil
1.	Klik menu rarangan	Menampilkan hasil rarangan Aksara Sunda	Berhasil
2.	Klik tombol <i>back</i>	Berpindah ke halaman sebelumnya	Berhasil

c. Pengujian *Interface* OCR Sunda-Latin

Tabel 4.3 dibawah ini merupakan pengujian *blackbox* testing pada menu OCR Sunda-Latin.

Tabel 4.3 Pengujian OCR Sunda-Latin

No.	Skenario Pengujian	Hasil yang diharapkan	Hasil
1.	Klik menu OCR Sunda – Latin.	Menampilkan halaman Upload gambar	Berhasil
2.	Klik tombol <i>back</i>	Berpindah ke halaman sebelumnya	Berhasil
3.	Klik tombol selanjutnya	Berpindah ke halaman Binerisasi	Berhasil

d. Pengujian *Interface Capture* gambar

Tabel 4.4 dibawah ini merupakan pengujian *blackbox* testing pada *interface Capture* Gambar.

Tabel 4.4 Pengujian *Interface capture* Gambar

No.	Skenario Pengujian	Hasil yang diharapkan	Hasil
1.	Klik tombol <i>upload</i> gambar dari galeri.	Berpindah ke internal <i>storage</i>	Berhasil
2.	Klik tombol <i>back</i> .	Berpindah ke halaman sebelumnya	Berhasil
3.	Klik tombol <i>upload</i> gambar dari kamera.	Memfoto gambar dari kamera	Berhasil

e. Pengujian *Interface* Binerisasi gambar

Tabel 4.5 dibawah ini merupakan pengujian *blackbox* testing pada *interface* Binerisasi Gambar.

Tabel 4.5 Pengujian *Interface* Binerisasi Gambar

No.	Skenario Pengujian	Hasil yang diharapkan	Hasil
1.	Klik Tombol Threshold	Menampilkan Nilai Threshold	Berhasil
2.	Klik tombol Selanjutnya	Berpindah ke halaman hasil pengenalan	Berhasil

f. Pengujian *Interface* Perbandingan Algoritma

Tabel 4.6 dibawah ini merupakan pengujian *blackbox testing* pada *interface* Perbandingan Algoritma.

Tabel 4.6 Pengujian *Interface* Perbandingan Algoritma

No.	Skenario Pengujian	Hasil yang diharapkan	Hasil
1.	Menampilkan hasil Waktu Algoritma	Menampilkan hasil Waktu Algoritma	Berhasil
2.	Menampilkan Aksara Latin hasil pengenalan	Menampilkan Aksara Latin hasil pengenalan	Berhasil

g. Pengujian *Interface* Menu Sejarah Aksara Sunda

Tabel 4.7 dibawah ini merupakan pengujian *blackbox testing* pada *Interface* menu sejarah aksara sunda.

Tabel 4.7 Pengujian *Interface* Menu Sejarah Aksara sunda

No.	Skenario Pengujian	Hasil yang diharapkan	Hasil
1.	Menampilkan hasil Waktu Algoritma	Menampilkan hasil Waktu Algoritma	Berhasil
2.	Menampilkan Aksara Latin hasil pengenalan	Menampilkan Aksara Latin hasil pengenalan	Berhasil

h. Pengujian *Interface* Menu Tentang

Tabel 4.8 dibawah ini merupakan pengujian *blackbox* testing pada *Interface* menu Tentang.

Tabel 4.8 Pengujian *Interface* Menu Tentang

No.	Skenario Pengujian	Hasil yang diharapkan	Hasil
1.	Klik Tombol Tentang	Menampilkan Menu Tentang	Berhasil
2.	Klik Tombol <i>back</i>	Kembali kehalaman sebelumnya	Berhasil

4.2 Pengujian Pengenalan Aksara Sunda

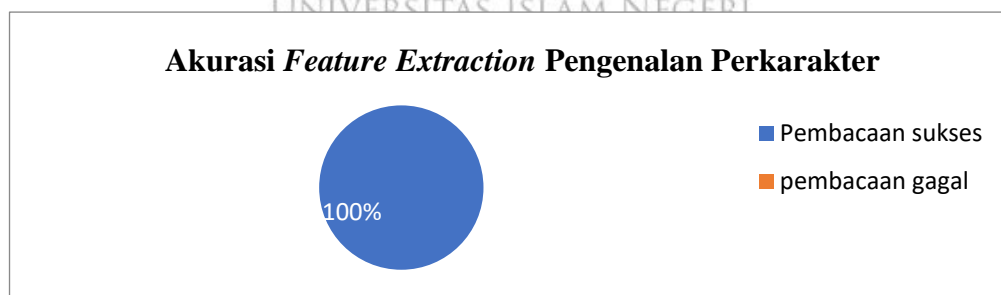
Tahap Pengujian Pengenalan Aksara Sunda merupakan tahap yang dilakukan untuk mengetahui kinerja dari sistem tersebut, dimana kinerja ini meliputi waktu proses dari algoritma *Template Matching* dan *Feature Extraction*, selain itu juga untuk mengetahui tingkat ketepatan dari hasil pengenalan kedua algoritma tersebut.

Dalam penelitian ini terdapat dua variasi pengujian yaitu pengujian aksara sunda dari tulisan digital (*digital writing*) dan pengujian aksara sunda dari tulisan tangan (*handwriting*), masing-masing pengujian dilakukan dengan variasi pengenalan aksara sunda perkarakter dan pengenalan aksara sunda perkata, *font* yang digunakan untuk melakukan pengujian *digital writing* adalah dengan menggunakan *font Sundanese.ttf* sedangkan untuk pengujian *handwriting* dilakukan dengan cara mengambil tiga data *sample* tulisan tangan yang berbeda.

4.2.1 Pengujian pengenalan aksara sunda tulisan digital (*digital writing*)

1. Pengujian Aksara sunda *digital writing* Perkarakter

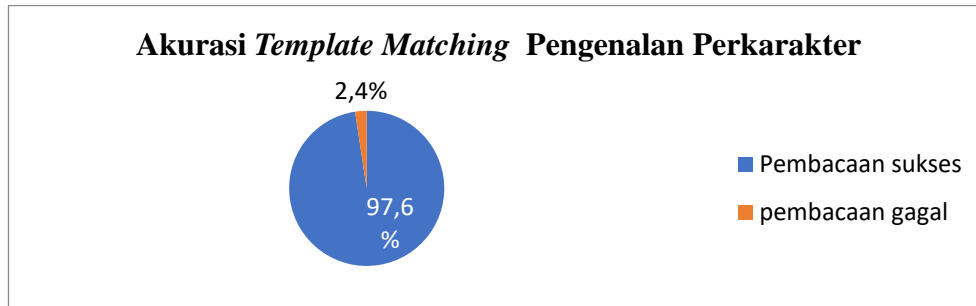
Pada pengujian ini aksara sunda akan diuji perkarakter, data akan diuji adalah aksara sunda konsonan, angka sunda, aksara vokal serta waktu proses yang dihasilkan dari pengenalan aksara sunda perkarakter, jumlah karakter yang diuji dalam pengujian ini berjumlah 42 karakter aksara sunda. Gambar 4.10 dan gambar 4.11 dibawah ini merupakan diagram dari akurasi *feature extraction* dan *template matching* dalam pengenalan aksara sunda perkarakter.



Gambar 4.10 Diagram Akurasi *Feature Extraction* Pengenalan Perkarakter

Dari hasil pengenalan pada *diagram* diatas bahwa *Feature extraction* mampu mengenali 42 karakter sunda secara tepat dengan nilai akurasi 100%.

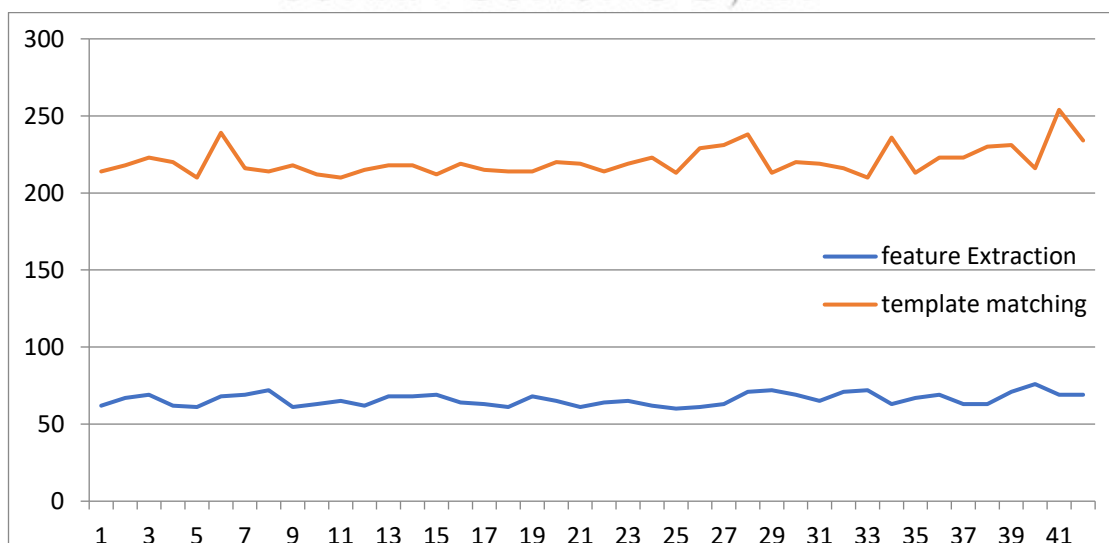
Pembacaan ini dilakukan dengan gambar input aksara sunda yang berformat .PNG atau .JPG.



Gambar 4.11 Diagram Akurasi *Template Matching* Pengenalan Perkarakter

Dari hasil pengenalan *Template Matching* pada gambar 4.11 di atas bahwa *Template Matching* dapat mengenali 41 karakter aksara sunda dari total 42 karakter, dengan nilai akurasi 97,6% dan nilai kesalahan 2,4%, kesalahan pembacaan terjadi pada saat pembacaan karakter // (la) terbaca sebagai angka sunda yaitu // (7) hal ini bisa terjadi karena kemiripan karakter sehingga terjadi salah pengenalan.

Gambar 4.12 dibawah ini merupakan perbandingan waktu antara *Template Matching* dengan *Feature Extraction*.



Gambar 4.12 Diagram Perbandingan waktu proses perkarakter

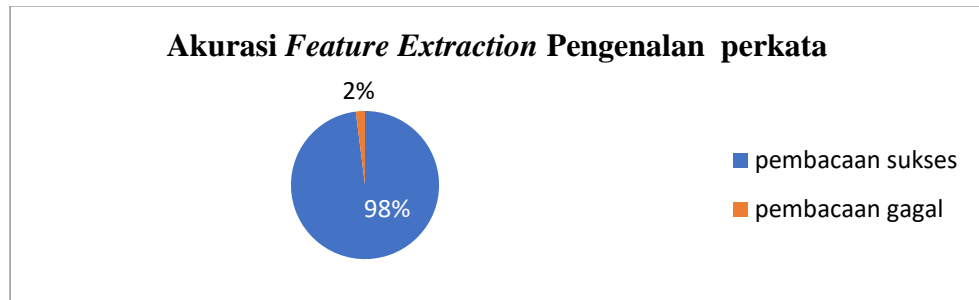
Gambar diatas merupakan diagram waktu dari *Feature Extraction* dan *Template Matching* berdasarkan pengenalan 42 karakter aksara sunda, dimana *Feature Extraction* memiliki nilai rata-rata waktu 66,023 ms, sedangkan *Template Matching* memiliki nilai rata-rata waktu 220,547 ms. *Feature Extraction* cenderung lebih cepat dibanding dengan *Template Matching*.

2. Pengujian Aksara sunda *digital writing* Perkata

Pada pengujian aksara sunda perkata ini menggunakan 30 *sample* kata sunda yang akan diuji, yaitu :

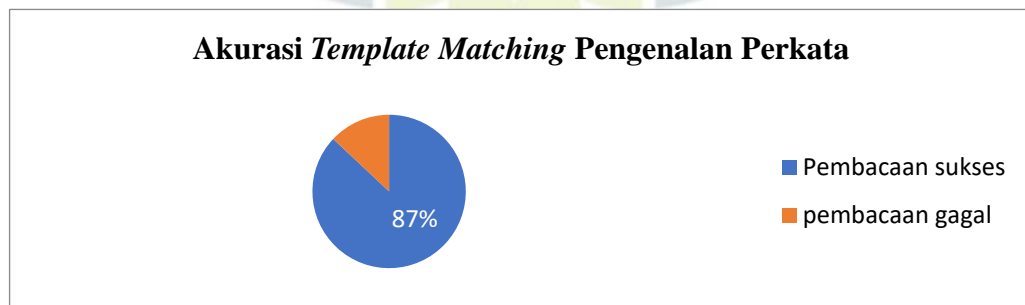
ᮊᮊᮊ (kamana)	ᮊᮊᮊ ₂ (alus)	ᮊᮊᮊ (timana)	ᮊᮊᮊᮊ (kumaha)	ᮊᮊᮊᮊ ₂ (babari)
ᮊᮊᮊᮊ ₂ (pisan)	ᮊᮊᮊᮊ ₂ (lalaki)	ᮊᮊᮊᮊᮊ ₂ (kasep)	ᮊᮊᮊᮊᮊ ₂ (muhun)	ᮊᮊᮊᮊᮊ ₂ (kanda)
ᮊᮊᮊᮊᮊ ₂ (dinda)	ᮊᮊᮊᮊ (saha)	ᮊᮊᮊᮊᮊ ₂ (loba)	ᮊᮊᮊᮊᮊ ₂ (atuh)	ᮊᮊᮊᮊᮊᮊ ₂ (sahaya)
ᮊᮊᮊᮊ ₂ (dina)	ᮊᮊᮊᮊᮊ ₂ (raraga)	ᮊᮊᮊᮊᮊᮊ ₂ (sasadu)	ᮊᮊᮊᮊᮊᮊ ₂ (naon)	ᮊᮊᮊᮊᮊᮊᮊ ₂ (isuk)
ᮊᮊᮊᮊᮊ ₂ (bapa)	ᮊᮊᮊᮊᮊᮊ ₂ (wasta)	ᮊᮊᮊᮊᮊᮊ ₂ (awewe)	ᮊᮊᮊᮊᮊᮊᮊ ₂ (janda)	ᮊᮊᮊᮊᮊᮊ ₂ (sore)
ᮊᮊᮊᮊᮊᮊᮊ ₂ (punte)	ᮊᮊᮊᮊᮊᮊ ₂ (kalem)	ᮊᮊᮊᮊᮊᮊ ₂ (sigana)	ᮊᮊᮊᮊᮊ ₂ (jalu)	ᮊᮊᮊᮊᮊ ₂ (rame)

Dari data kata aksara sunda diatas diuji tingkat akurasi serta tingkat kecepatan waktunya, hasil dari pengujian tersebut terdapat pada diagram dibawah ini.



Gambar 4.13 Diagram Akurasi *Feature Extraction* Pengenalan Perkata

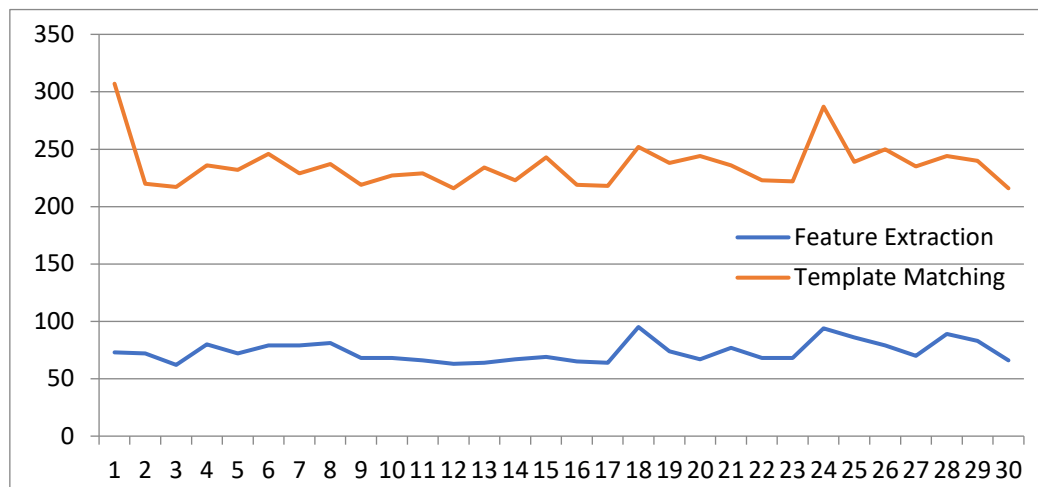
Pada gambar 4.13 diatas merupakan hasil dari algoritma *feature extraction* untuk pengenalan perkata, memiliki akurasi sebesar 98%, terdapat beberapa kesalahan kata yang dikenali diantaranya kata ᮊᮊᮊᮊ (babari) dikenali sebagai kata ᮊᮊᮊᮊ (babare), ᮊᮊᮊ (loba) dikenali ᮊᮊᮊ (laneba), dan ᮊᮊᮊ (atuh) dikenali ᮊᮊᮊ (atua).



Gambar 4.14 Diagram Akurasi *Template Matching* Pengenalan Perkata

Gambar 4.14 diatas merupakan Data pembacaan dari Algoritma *Template Matching* dimana memiliki nilai akurasi sebesar 87%, dan nilai kesalahan sebesar 13%, kesalahan pengenalan pada algoritma *template matching* diantaranya pada pengenalan kata ᮊᮊᮊᮊ (kumaha) menjadi ᮊᮊᮊᮊ (pamaha), ᮊᮊᮊᮊ (babari) menjadi ᮊᮊᮊᮊ (babare), ᮊᮊᮊᮊ (kasep) menjadi ᮊᮊᮊᮊ (kasangp), ᮊᮊᮊᮊ (atuh) menjadi ᮊᮊᮊᮊ (atusyi), ᮊᮊᮊᮊ (sore) menjadi ᮊᮊᮊᮊ (sori), ᮊᮊᮊᮊ (kalem) menjadi ᮊᮊᮊᮊ (kalangm).

Gambar 4.15 dibawah ini merupakan perbandingan waktu antara *Template Matching* dengan *Feature Extraction* dalam pengenalan aksara perkata.



Gambar 4.15 Perbandingan waktu proses Pengenalan Perkata

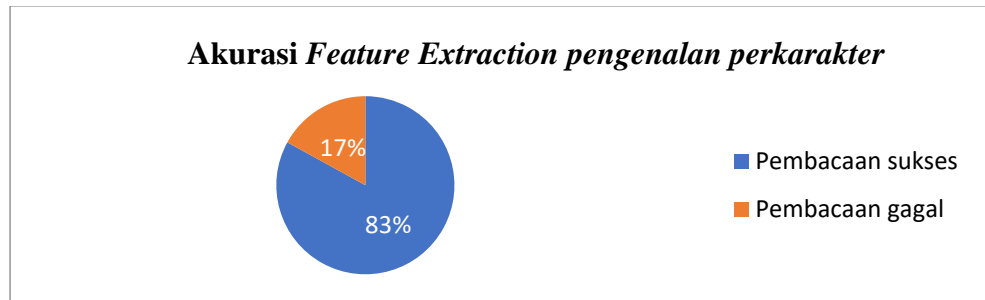
Pada gambar 4.15 diatas merupakan perbandingan waktu proses *Template Matching* dengan *Feature Extraction* dalam pengenalan aksara sunda perkata, rata-rata waktu proses *Feature Extraction* adalah 73,6 ms, sedangkan waktu rata-rata *Template Matching* adalah 236 ms. Dapat dikatakan proses *Feature Extraction* lebih cepat daripada proses *Template Matching*.

4.2.2 Pengujian pengenalan aksara sunda tulisan tangan (*handwriting*)

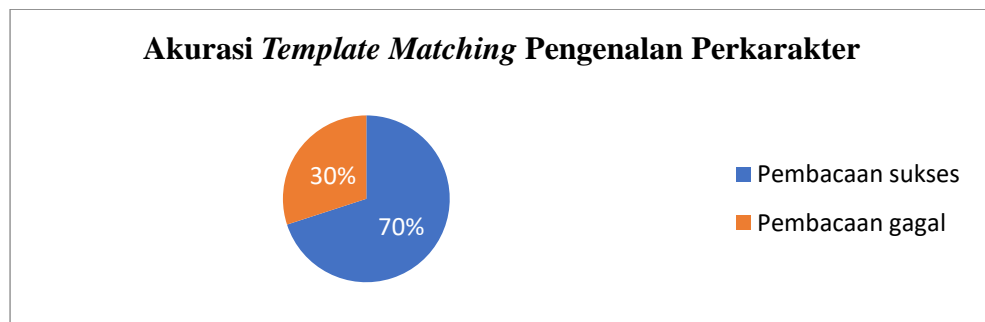
Pengujian *handwriting* dilakukan dengan menggunakan tiga data *sample* dari tulisan tangan yang berbeda, sepuluh data untuk pengenalan perkarakter dan sepuluh data untuk pengenalan perkata.

1. Pengujian Aksara sunda *handwriting* Perkarakter.

Gambar diagram 4.16 dan Gambar 4.17 merupakan hasil dari pengujian *handwriting* aksara sunda perkarakter.



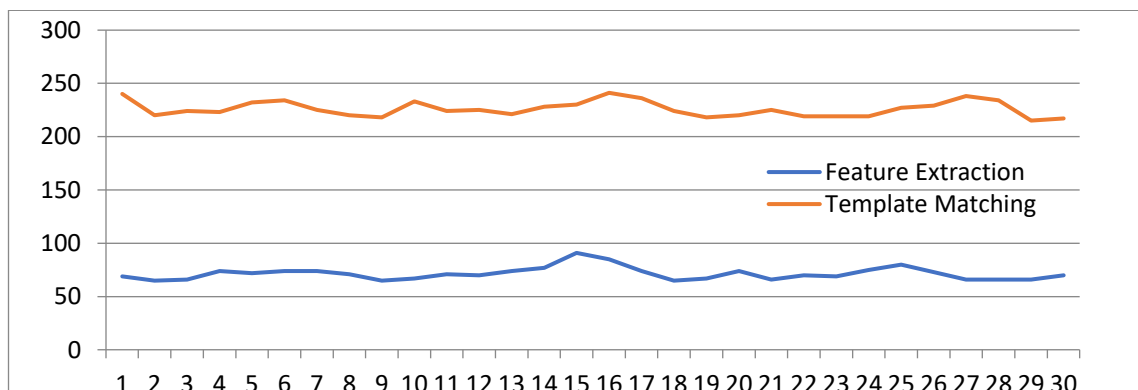
Gambar 4.16 Diagram Akurasi *Feature Extraction* pengenalan Karakter



Gambar 4.17 Diagram Akurasi *Template Matching* pengenalan Karakter

Dari hasil pengujian perkarakter, dapat dilihat algoritma *feature extraction* dapat membaca 83% karakter dengan nilai pembacaan gagal 17%, sedangkan untuk algoritma *template matching* memiliki nilai akurasi 70% dengan nilai pembacaan gagal 30%.

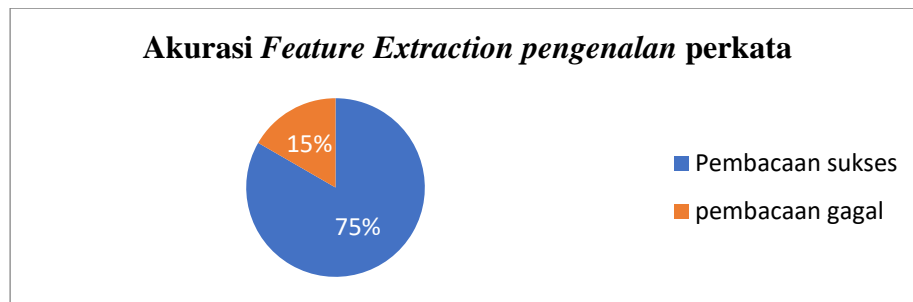
Gambar 4.18 dibawah ini merupakan perbandingan waktu antara *Template Matching* dengan *Feature Extraction* dalam pengenalan tulisan tangan karakter aksara sunda.



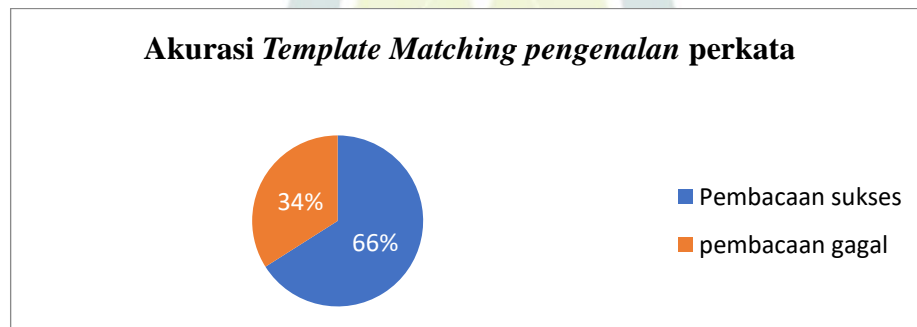
Gambar 4.18 Waktu proses Pengenalan Perkarakter

Waktu proses rata-rata untuk pengenalan perkarakter algoritma *feature extraction* adalah 71,5 ms, sedangkan untuk algoritma *Template matching* rata-rata waktu proses pengenalan aksara sunda perkarakter adalah 226 ms.

2. Pengujian Aksara sunda *handwriting* Perkata.



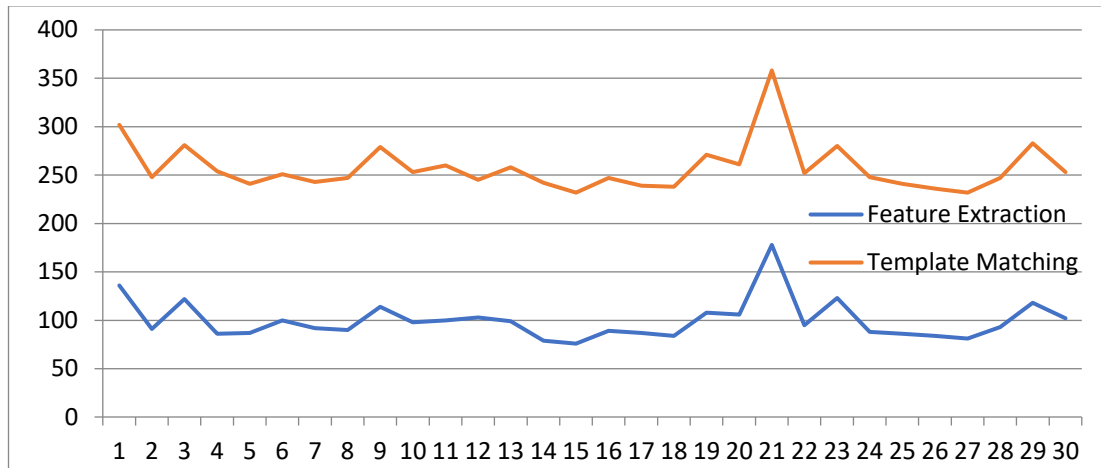
Gambar 4.19 Diagram akurasi *Feature Extraction* pengenalan perkata



Gambar 4.20 Diagram akurasi *Template Matching* pengenalan perkata

hasil pengenalan *Handwriting* perkata untuk algoritma *feature extraction* memiliki tingkat akurasi 75% dan memiliki tingkat kegagalan pembacaan sebesar 15%, sedangkan untuk algoritma *Template Matching* memiliki Akurasi sebesar 66% dengan kegagalan pembacaan 34%.

Gambar 4.21 dibawah ini merupakan perbandingan waktu antara *Template Matching* dengan *Feature Extraction* dalam pengenalan tulisan tangan kata dalam aksara sunda.



Gambar 4.21 Waktu proses Pengenalan Perkata

Waktu proses rata-rata untuk pengenalan perkata algoritma *feature extraction* adalah 99,83 ms, sedangkan untuk algoritma *Template matching* rata-rata waktu proses pengenalan aksara sunda perkata adalah 257,4 ms.