BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Seiring dengan perkembangan teknologi, fungsi indra pada manusia terutama penglihatan dan pendengaran dapat ditiru oleh sebuah sistem buatan. Peranan penting dalam pembuatan sistem tersebut ialah pengenalan pola, diantaranya penelitian yang dapat terapkan seperti pengenalan sidik jari, pengenalan wajah, pengenalan iris mata, pengenalan telapak tangan dan pengenalan lainnya yang dapat dijadikan identitas dengan menggunakan komputer sebagai alat bantu. Seperti halnya dalam konteks ini berfokus terhadap pengenalan huruf aksara Lontara. Dengan adanya penelitian ini diharapkan generasi selanjutnya mampu membawa bahasa khas daerah mereka khususnya bahasa Bugis/Makassar yang mempunyai aksara sendiri yaitu lontara ke dunia digital. Pengembangan sistem dapat menerjemahkan huruf lontara agar Masyarakat mengerti dan mampu membaca huruf lontara. Variasi pada pola aksara lontara terdiri dari dua puluh tiga karakter. Karakter aksara lontara bersifat kompleks disebabkan karena masing-masing dari pola karakter memiliki banyak kemiripan.

*Opticial Character Recognit*ion (OCR) adalah sistem komputer pengenalan karakter yang dapat membaca huruf, baik yang berasal dari sebuah pencetak (printer atau mesin ketik) maupun yang berasal dari tulisan tangan. Dari peneliti sebelumnya proses pengenalan OCR menggunakan dua metode yaitu mencocokkan matriks *(matric matching*) atau ekstraksi fitur (*feature extraction*). Terbentuknya

sistem pengenalan karakter ini diharapkan mampu mempermudah digitalisasi informasi dan pengetahuan, misalnya dalam pembuatan koleksi pustaka digital, koleksi sastra kuno digital, dan lain-lain. Dengan demikian metode yang akan digunakan pada penelitian ini yaitu *Opticial Character Recognition* (OCR). Berdasarkan hal tersebut sehingga diangkat topik "Implementasi Aplikasi *Optical Character Recognition* Pengenalan Tulisan Aksara Lontara Berbasis *Mobile*".

B. Rumusan Masalah

Berdasarkan latar belakang yang diuraikan di atas, permasalahan yang dapat di rumuskan yaitu bagaimana membuat sistem aplikasi *optical character recognition* pengenalan tulisan aksara lontara berbasis mobile?

C. Tujuan Penelitian

Tujuan dari penelitian ini ialah:

1. Menerapkan teknologi OCR dalam pengenalan tulisan aksara Lontara.
2. Mengetahui tingkat akurasi dalam mengenali pola yang benar

D. Batasan Masalah

Adapun batasan masalah yang ada dalam penelitian ini ialah :

1. Aplikasi berbasis android.
2. Huruf tulisan adalah aksara lontara.
3. Tampilan citra aksara lontara jelas.
4. Tipe file citra adalah jpg, png, bitmap *picture*.

E. Manfaat Penelitian

Adapun manfaat penelitian yang diharapkan dari hasil penelitian ini adalah sebagai berikut :

1. Menambah pengetahuan penulis tentang teknologi OCR untuk mengenali pola.
2. Dapat mengimplementasikan ilmu yang di dapat di bangku kuliah dengan membuat aplikasi berbasis *mobile*.
3. Menambah ilmu pengetahuan dan informasi bagi mahasiswa teknik informatika Universitas Muhammadiyah Parepare tentang teknologi OCR.

F. Sistematika Penulisan

Struktur penulisan penelitian ini adalah :

BAB I : PENDAHULUAN

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan dari pembuatan proposal ini.

BAB II : TINJAUAN PUSTAKA

Berisi tentang penelitian kajian teori, kajian hasil penelitian dan kerangka pikir yang mendukung pembuatan skripsi ini.

BAB III : METODE PENELITIAN

Berisi tentang letak lokasi serta waktu, jenis penelitian, metode pengumpulan data, alat dan bahan penelitian, tahap penelitian, metode pengujian, dan anlisa sistem.

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Berisi tentang gambaran umum, analisis, meliputi pendefenisisan dan pemodelan sistem dalam bentuk *use case, class* diagram, *sequence* diagram serta disain *database*, dan pengujian sistem dengan *whitbox* dan *blackbox.*

BAB V : PENUTUP

Berisi kesimpulan yang dapat di ambil dari penulisan akhir ini dan saran-saran pengembangannya.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. *Opticial Character Recognition* (OCR)

(Revi Kuswanar, 2019). *Opticial Character Recognition* (OCR) adalah sistem komputer pengenalan karakter yang dapat membaca huruf, baik yang berasal dari sebuah pencetak (printer atau mesin ketik) maupun yang berasal dari tulisan tangan.



Gambar 2. 1. Proses Umum OCR

a. Preprosessing

Preprosessing adalah tahap pertama yang dilakukan sebelum data citra digunakan untuk proses pengenalan. Proses ini merupakan tahap yang terpenting karena memiliki pengaruh besar terhadap keberhasilan suatu sistem pengenalan. Tujuannya menyeragamkan bentuk maupun sifat citra yang akan digunakan dalam proses nantinya, sehingga tidak menimbulkan perbedaan yang besar ketika proses pengenalan berjalan. Berikut ini adalah tahap preprosesing yang akan dilakukan.

1) Binerisasi adalah tahap dimana file citra digital dikonversi menjadi citra biner. Seperti dijelaskan pada subbab citra digital bahwa citra biner merupakan citra yang hanya memiliki dua derajat keabuan, yaitu hitam dan putih. Pixel-pixel objek bernilai 1 dan pixel-pixel latar belakang bernilai 0.

Dengan kata lain pada citra biner, latar belakang berwarna putih sedangkan objek berwarna hitam.

2) Metode Otsu (*Otsu Thresholding*) Metode Otsu merupakan metode perhitung memilih batas ambang T secara otomatis berdasarkan citra masukan. Pendekatan yang digunakan oleh metode Otsu adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis diskriminan merupakan teknik statistika yang dipergunakan untuk mengklasifikasikan suatu individu atau observasi ke dalam suatu kelas atau kelompok berdasarkan sekumpulan variabel-variabel.

b. Segmentasi

Segmentasi adalah proses memisahkan area pengamatan (*region*) pada tiap karakter yang dideteksi. Segmentasi citra biner bertujuan untuk mengelompokkan pixel-pixel objek menjadi wilayah yang merepresentasikan suatu objek (Munir, 2010). Batas antara objek dengan latar belakang akan terlihat jelas pada citra biner. *Pixel* objek akan terlihat berwarna putih sedangkan *pixel* latar belakang berwarna hitam. Secara matematis dapat dijelaskan sebelum proses segmentasi dilakukan aka nada proses *thresholding*, dimana nilai elemen-elemen dalam sebuah matriks citra akan diubah dengan nilai 0 dan 255 atau dengan kata lain diubah menjadi warna hitam dan putih. Selanjutnya pada proses segmentasi, nilai 0 yang menjadi representasi dari warna putih akan diubah menjadi 1 yang merupakan representasi dari warna hitam. Begitupun dengan nilai 255 representasi dari warna hitam akan diubah menjadi 0 yang merupakan representasi nilai putih.

c. Normalisasi

Normalisasi merupakan salah satu tahap preprosessing, dimana pada tahap ini ukuran citra data uji akan disesuai dengan citra data latih yang telah diolah sebelumnya. Jadi, data latih yang memiliki ukuran matriks 30x50 akan menjadi patokan ukuran dari data uji yang akan digunakan. Sebelum proses perhitungan berlangsung ukuran matriks data uji telah diubah menjadi 30x50, dalam hal ini memiliki ukuran sama dengan data latih yang telah disiapkan. Pada dasarnya proses ini disesuaikan dengan kebutuhan pada proses pengenalan yang digunakan.

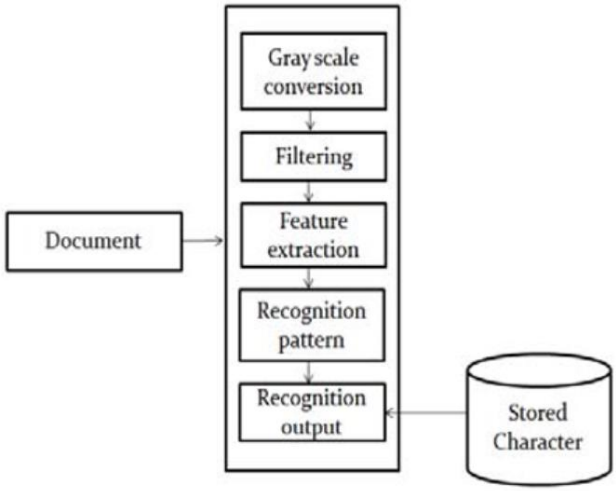
d. Ektraksi Fitur

Ektraksi fitur merupakan suatu metode untuk mendapatkan karakteristik dari suatu citra (dalam hal ini citra tersebut merupakan suatu karakter berupa huruf 16 aksara). Dengan ektraksi fitur maka citra yang satu dengan yang lain dapat dibedakan denganmemperhatikan ciri yang terdapat pada citra itu sendiri. Ekstraksi fitur yang akan digunakan pada penelitian ini yakni susunan elemen - elemen matriks ordo 30x50 yang telah bernilai biner. Selanjutnya setiap matriks dari karakter tersebut di kumpulkan dalam suatu tabel yang memiliki banyak cell.

e. Recognition

Recognition merupakan proses untuk mengenali karakter yang diamati dengan cara membandingkan ciri-ciri karakter yang diperoleh dengan ciri-ciri karakter yang ada pada file data latih. Pada proses ini akan digunakan algoritma Manhattan yang akan dijelaskan pada penjelasan berikutnya.

Character recognition juga diklasifikasikan kedalam dua tipe berdasarkan metode tulisannya, antara lain: *Optical Character Recognition* (OCR) dan *Handwritten Character Recognition* (HCR). Dimana akurasi pada HCR biasanya masih lebih rendah dikarnakan besarnya perbedaan bentuk dan tipe tulisan. Perbedaan karakter dalam Bahasa juga berpengaruh besar, contohnya: tulisan kanji mandarin, jepang, dan lainnya. Dalam proses OCR berikut ini gambaran sistem yang akan dilakukan:



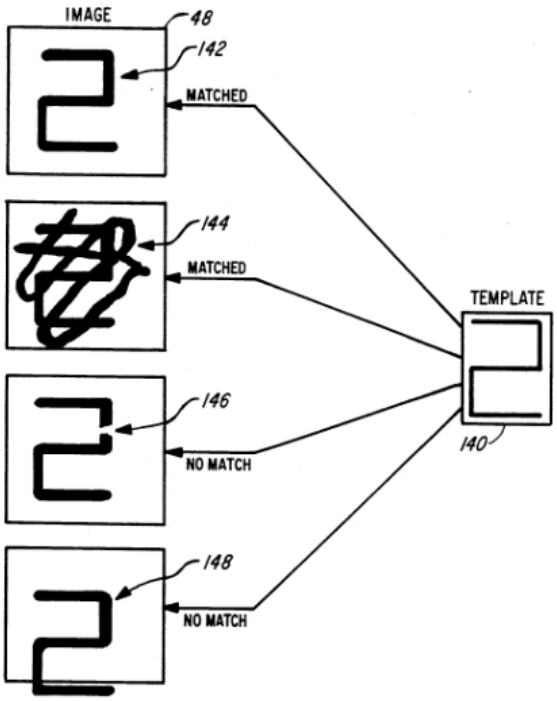
Gambar 2. 2. Sistem Blok Diagram

Offline* dan *Online Character Recognition

Offline character recognition system men-generate dokumen terlebih dahulu, melakukan digitalisasi, dan menyimpannya kedalam computer, sebelum memprosesnya. Dimana untuk *online character recognition system*, *character* langsung di proses selama dalam pembuatan. Faktor eksternal seperti kecepatan menulis berpengaruh pada kasus offline system. *Offline* ataupun *Online system* dapat diterapkan untuk optical maupun *handwritten characters recognition*

Template Matching

Template matching dalam OCR adalah paten yang dimiliki oleh Hin-Leong Tan (Tan, 1991). Dalam *template matching*, pixel masing-masing gambar dicocokkan dengan *template* yang sudah disediakan, karakter yang memiliki kecocokkan tertinggi dianggap sebagai karakternya.



Gambar 2. 3. Simulasi Template Matching

2. Aksara Lontara

(Syahruddin Wahid, dkk. 2020). Lontara berasal dari kata 'lontar' yang merupakan spesies flora endemik di Sulawesi Selatan. Lontara menjadi aksara tradisional masyarakat Bugis dan Makassar. Awalnya, aksara Lontara diciptakan oleh Daeng Pamette, seorang syahbandar sekaligus tumailalang atau menteri urusan dalam dan luar negeri pada Karaeng Tumapakrisi Kallonna, Raja Gowa ke 9 yang memerintahkan pada abad ke 14. Aksara digunakan untuk menulis pesan atau dokumen penting lainnya, jauh sebelum kertas ditemukan. Aksara Lontara

mempunyai peranan yang sangat penting dalam kehidupan masyarakat orang Bugis. Karena, aksara Lontara mengandung nilai-nilai budaya yang tinggi sebagai pedoman hidup dan kehidupan masyarakat Bugis. Aksara Lontara terdiri dari 23 huruf Lontara Bugis dan 19 huruf Lontara Makassar. Perbedaan, aksara Lontara Bugis dan Makassar, yaitu Lontara Bugis dikenal dengan huruf nga', mpa', nca', dan nra'. Sedangkan, Lontara Makassar huruf tersebut tidak ada. Terdapat 23 aksara dasar dalam aksara Lontara, sebagaimana berikut:

Tabel 2. 1. Aksara Lontara

Ka	Ga	nga	ngka
			
			
Pa	**Ba**	**ma**	**mpa**
			
			
Ta	**Da**	**na**	**nra**
			
			
Ca	**Ja**	**nya**	**nca**
			
			
ya	**ra**	**la**	**wa**
			
			
sa	**a**	**ha**	

3. Android

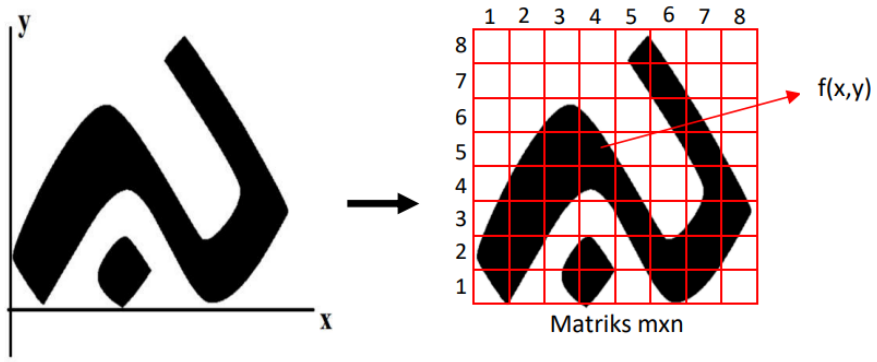
(Abdul Kadir. 2020). Android adalah sistem operasi berbasis Linux yang dipergunakan sebagai pengelola sumber daya perangkat keras, baik untuk ponsel,

smartphone dan juga PC tablet. Secara umum Android adalah platform yang terbuka (Open Source) bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang digunakan oleh berbagai piranti bergerak. Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android. Semenjak kehadirannya pada 9 Maret 2009, Android telah hadir dengan versi 1.1, yaitu sistem operasi yang sudah dilengkapi dengan pembaruan estetis pada aplikasinya, seperti jam alarm, voice search, pengiriman pesan dengan Gmail, dan pemberitahuan email. Pada perkembangannya, sistem operasi Android telah mengalami beberapa perubahan dan perbaikan. Dan yang paling menarik adalah versi keluaran Android yang diberi nama seperti nama-nama makanan.

4. Citra Digital

(Pamungkas. 2017). Image atau citra merupakan gambar pada bidang dwimatra (dua dimensi). Sedangkan dilihat dari sudut pandang matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dwimatra. Citra terdiri dari dua jenis jika dilihat dari segi sinyal penyusun, yaitu citra kontinu dan citra diskrit. Citra kontinu berasal dari sistem optik yang menerima sinyal analog. Nilai intensitas cahaya pada citra kontinu memiliki range antara 0 s.d ~. Alat akuisisi citra kontinu antara lain mata manusia dan kamera analog. Karena sifatnya yang kontinu jenis citra ini tidak dapat disimpan atau diolah dalam komputer. Sedangkan citra diskrit berasal dari citra yang terbentuk dari sinyal digital. Nilai intensitas cahaya pada citra distrit bergantung pada nilai bit yang menyusunnya. Representasi citra

dari fungsi intensitas cahaya menjadi nilai-nilai diskrit disebut sebagai digitalisasi. Citra yang dihasilkan inilah disebut citra digital.Dalam bidang dua dimensi, citra dibentuk oleh sekumpulan picture element (pixel) yang memiliki dua informasi penting yaitu koordinat piksel (x,y) dan nilai intensitas piksel f(x,y). Gambar di bawah adalah ilustrasi citra digital dalam representasi menjadi nilai-nilai diskrit.



Gambar 2. 4. Ilustrasi Citra Digital

Ada tiga jenis citra yang umum digunakan dalam pemrosesan citra, yaitu citra berwarna, citra berskala keabuan dan citra biner. Citra berwarna atau biasa disebut Citra RGB merupakan citra yang tersusun oleh tiga komponen warna seperti singkatan sebutannya yaitu R yang berarti merah, G yang berarti hijau, dan B yang berarti biru. Pada citra RGB 24-bit, masing-masing komponen warna memiliki nilai intensitas piksel dengan kedalaman bit sebesar 8-bit memiliki nilai berkisar antara 0 sampai dengan 255. Setiap piksel pada citra RGB memiliki nilai intensitas yang merupakan kombinasi dari nilai R, G, dan B. Variasi warna yang dapat disajikan dalam piksel pada citra RGB adalah sekitar 256 x 256 x 256 atau 16777216.

Citra berskala keabuan atau grayscale tersusun dari gradasi warna hitam dan putih yang menghasilkan efek warna abu-abu.Sebuah citra skala keabuan mempunyai jumlah tingkat keabuan yang spesifik. Untuk citra skala keabuan 8 bit

bisa merepresentasikan 28 – 1 = 255 intensitas. Dalam hal ini, intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih. Citra biner adalah citra yang setiap piksel hanya dinyatakan dengan sebuah nilai dari dua kemungkinan (yaitu nilai 0 dan 1). Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih.

5. *Visual Studio Code*

Visual Studio Code (VSCode) adalah editor kode sumber populer dan serbaguna yang dikembangkan oleh Microsoft. Visual Studio Code awalnya dirilis oleh Microsoft pada tahun 2015 sebagai alternatif lintas platform, sumber terbuka, dan ringan untuk Visual Studio IDE yang berfokus pada Windows. Microsoft merilis VS Code untuk melayani khalayak pengembang yang lebih luas yang akrab dengan berbagai sistem operasi, termasuk Windows, macOS, dan Linux.

Salah satu kekuatannya yang menonjol adalah sistem ekstensinya yang kuat, memungkinkan pengguna untuk menyesuaikan dan meningkatkan pengalaman coding mereka dengan beragam ekstensi yang tersedia di Visual Studio Code *Marketplace*. Ekstensibilitas ini membuat VSCode cocok untuk beragam tugas pengembangan, mulai dari pengembangan web hingga ilmu data.

Antarmuka pengguna *Visual Studio Code* bersih dan intuitif, memberikan pengalaman yang lancar baik bagi pemula maupun pengembang berpengalaman. Fitur lingkungan pengembangan terintegrasi (IDE) mencakup penyorotan sintaksis, penyelesaian kode, dan integrasi Git, yang menyederhanakan proses pengkodean. Selain itu, VSCode

mendukung banyak bahasa dan kerangka kerja, dengan alat debugging bawaan yang menyederhanakan identifikasi dan penyelesaian kesalahan pengkodean. Kompatibilitas multi-platformnya, mendukung Windows, macOS, dan Linux, semakin meningkatkan daya tariknya, membuatnya dapat diakses oleh komunitas pengembang yang luas.

Fitur luar biasa dari *Visual Studio Code* adalah fokusnya pada kolaborasi dan produktivitas. Ekstensi Live Share memungkinkan sesi pengkodean kolaboratif secara real-time, memungkinkan pengembang untuk bekerja sama dalam basis kode yang sama, meningkatkan kerja tim dan mengurangi waktu pengembangan. Selain itu, terminal terintegrasi dan kemampuan debugging yang kuat berkontribusi pada alur kerja pengkodean yang efisien. Dengan komitmen terhadap pembaruan dan peningkatan rutin, *Visual Studio Code* terus berkembang, mempertahankan statusnya sebagai pilihan utama bagi pengembang yang mencari editor kode yang andal dan kaya fitur.

Popularitas VS Code dengan cepat tumbuh berkat ekosistem ekstensi besar-besaran yang didukung oleh pengembang yang menyumbangkan fungsionalitas tambahan untuk berbagai alat, bahasa pemrograman, dan layanan. Selain ekstensinya, VS Code juga menarik perhatian para pengembang karena sifatnya yang serbaguna dan ringan, menjadikannya solusi yang layak untuk pengembangan web, ilmu data, dan banyak tugas pengembangan lainnya. VS Code terus menjadi pilihan utama bagi pengembang perangkat lunak dan web yang mencari editor kode multi-

bahasa dan lintas platform untuk menulis, melakukan debug, dan mengelola kode secara efisien serta meningkatkan produktivitas.

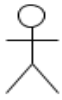
6. *UML (Unified Modelling Language)*

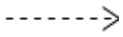
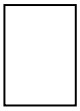
N, Sora (2020), *Unified Modelling Language (UML)* adalah sekumpulan alat yang digunakan untuk melakukan abstraksiterhadap sebuah sistem atau perangkat lunak berbasis objek. *UML* merupakan singkatan dari *Unified Modeling Language*. *UML* juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena developer harus melakukan penelusuran dan mempelajari kode program *UML* juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu developer ke developer lainya. Tidak hanya antar developer terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya *UML*. Dalam *UML* sendiri terdapat beberapa diagram yaitu :

a. *Use Case Diagram*

Adapun simbol-simbol *Use Case Diagram* antara lain :

Tabel 2. 2. Simbol *Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna *Main*kan ketika berinteraksi dengan *Use Case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).

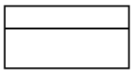
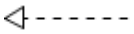
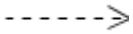
NO	GAMBAR	NAMA	KETERANGAN
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *Use Case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *Use Case* target memperluas perilaku dari *Use Case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. *Class Diagram*

Adapun simbol-simbol *Class Diagram* antara lain :

Tabel 2. 3. Simbol *Class* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan

NO	GAMBAR	NAMA	KETERANGAN
			struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

c. *Sequence Diagram*

Adapun simbol-simbol *Sequence* Diagram antara lain :

Tabel 2. 4. Simbol *Sequence* Diagram

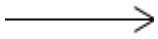
NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-

			informasi tentang aktifitas yang terjadi

d. *StateChart Diagram*

Adapun simbol-simbol *StateChart Diagram* antara lain :

Tabel 2. 5. Simbol *StateChart* Diagram

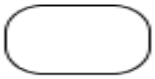
NO	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

e. *Activity Diagram*

Adapun simbol-simbol *Activity Diagram* antara lain :

Tabel 2. 6. Simbol *Activity* Diagram

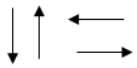
NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain

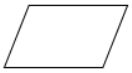
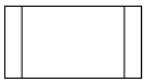
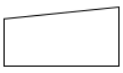
NO	GAMBAR	NAMA	KETERANGAN
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

7. *Flowchart*

Arifianto Rahmad (2017), *Flowchart* adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu *Flowchart* :

Tabel 2. 7. Simbol *Flowchart*

Simbol	Nama dan Fungsinya
	***Terminal Point Symbol / Simbol Titik****Terminal Menu*njukkan permulaan (start) atau akhir (stop) dari suatu proses.
	***Flow Direction Symbol* / Simbol Arus** adalah simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain (*connecting line*). Simbol ini juga berfungsi untuk *Menu*njukkan garis alir dari proses.
	***Processing Symbol* / Simbol** Proses digunakan untuk *Menu*njukkan kegiatan yang dilakukan oleh komputer. Pada bidang industri (proses produksi barang), simbol ini menggambarkan kegiatan inspeksi atau yang biasa dikenal dengan simbol inspeksi

Simbol	Nama dan Fungsinya
	***Decision Symbol* / Simbol** Keputusan merupakan simbol yang digunakan untuk memilih proses atau keputusan berdasarkan kondisi yang ada. Simbol ini biasanya ditemui pada *Flowchart* program.
	**I*nput-Output* / Simbol Keluar**-Masuk *Menu*njukkan proses *input-output* yang terjadi tanpa bergantung dari jenis peralatannya.
	***Predefined Process* / Simbol Proses** *Terdefinisi*merupakan simbol yang digunakan untuk *Menu*njukkan pelaksanaan suatu bagian prosedur (*sub*-proses).
	***Connector (On-page)*** Simbol ini fungsinya adalah untuk menyederhanakan hubungan antar simbol yang letaknya berjauhan atau rumit bila dihubungkan dengan garis dalam satu halaman
	***Connector (Off-page)*** Sama seperti *on-page connector*, hanya saya simbol ini digunakan untuk menghubungkan simbol dalam halaman berbeda. label dari simbol ini dapat menggunakan huruf atau angka
	***Preparation Symbol* / Simbol** Persiapan merupakan simbol yang digunakan untuk mempersiapkan penyimpanan di dalam storage.
	***Manual Input Symbol*** digunakan untuk *Menu*njukkan input data secara manual menggunakan online keyboard.
	***Manual Operation Symbol* / Simbol Kegiatan** Manual digunakan untuk *Menu*njukkan kegiatan/proses yang tidak dilakukan oleh komputer.
	***Document Symbol*** Jika Anda menemukan simbol ini artinya input berasal dari dokumen dalam bentuk kertas, atau output yang perlu dicetak di atas kertas.

Simbol	Nama dan Fungsinya
	***Multiple Documents*** sama seperti *document symbol* hanya saja dokumen yg digunakan lebih dari satu dalam simbol ini
	***Display Symbol*** adalah simbol yang menyatakan penggunaan peralatan output, seperti layar monitor, printer, plotter dan lain sebagainya
	***Delay Symbol*** sesuai dengan namanya digunakan untuk *Menu*njukkan proses delay (*Menu*nggu) yang perlu dilakukan. Seperti *Menu*nggu surat untuk diarsipkan

B. Kajian Hasil Penelitian

Penelitian relevan berisikan penelitian yang terdahulu kemudian dicari bahan penelitian yang ada dan relevan dengan penelitian ini karena sangat berguna untuk mendukung kajian teori yang dikemukakan, sehingga dapat digunakan sebagai landasan pada penyusunan kerangka pikir. Beberapa hasil penelitian yang relevan dengan penelitian ini adalah penelitian yang dilakukan oleh:

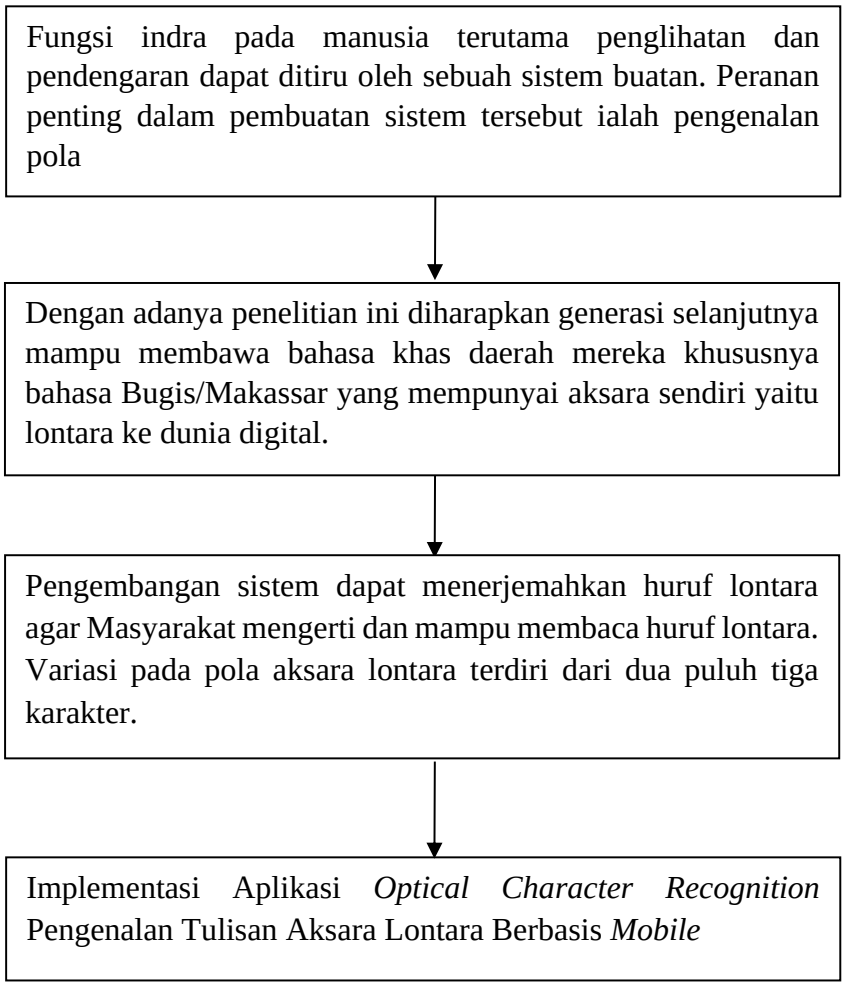
1. Suryo Hartanto, dkk (2018) dengan optical character recognition menggunakan algoritmatemplate matching correlation.Denga menggunakan metode templatematching correlation pada penelitian proses pengenalan, citra masukan dibuat dengan format bmp atau jpg lalu diolah pada tahapan

praproses, yang meliputi binerisasi, segmentasi, dan normalisasigambar. Rata-rata tingkat keberhasilan pengenalan yang dihasilkan oleh sistem ini adalah 92,90%.Hasil akhir menunjukkan bahwa penggunaan metode template matchingcorrelation cukup untukmembangun sebuah sistem OCR dengan akurasi yang baik efektif.

2. Oki Fredian (2016) judul penelitian yaitu aplikasi pengenalan plat nomor kendaraan dengan metode optical character recognition (OCR). Penelitian dilakukan terhadap beberapa sampel citra plat nomor yang berjumlah 10 buah dengan masing- masing dari kota yang berbeda di wilayah Surakarta. Semua sampel tersebut melalui tahap proses akusisi hingga proses identifikasi dengan hasil rata-rata tingkat keberhasilan 96,25%. Berdasarkan 10 sampel hanya 8 sampel citra yang berhasil dengan tingkat keberhasilan 100% dan 2 sampel citra terdapat sedikit kesalahan dengan tingkat keberhasilan 75% dan 87,5%. Sampel citra plat nomor yang berjumlah 8 buah tersebut berhasil dikarenakan karakter pola citra plat nomor dengan citra template matching ketika melalui proses OCR perhitungan nilai binernya hampir sama. Sedangkan 2 buah citra plat nomor yang terdapat kesalahan karena ketidaksamaan nilai biner karakter pola citra plat nomor dengan citra template matching sehingga, diambil pola citra lain yang nilai binernya terdekat

C. Kerangka Pikir

Untuk lebih memperjelas kerangka pikir maka digambarkan dalam bentuk diagram sebagai berikut :



Gambar 2. 5. Kerangka Pikir

BAB III

METODE PENELITIAN

A. Lokasi dan Waktu Penelitian

Lokasi penelitian dilakukan di Universitas Muhammadiyah Parepare. Waktu yang dipergunakan untuk melakukan penelitian dimulai pada bulan November 2023 – Januari 2024.

Tabel 3. 1. Jadwal Pelaksanaan Penelitian

No.	Uraian Kegiatan	Bulan 2023 - 2024		
		November	Desember	Januari
1	Studi Literatur			
2	Pembuatan Aplikasi			
3	Pengujian Aplikasi			
4	Hasil Pengujian			

B. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian terapan mengenali pola aksara lontara dengan menggunakan metode *Optional Character Recognition*. Penelitian terapan adalah salah satu jenis penelitian yang bertujuan untuk memberikan solusi atas permasalahan tertentu secara praktis.

C. Metode Pengumpulan Data

Metode yang digunakan untuk memperoleh data yang berkaitan dengan penelitian ini adalah observasi atau pengamatan dan telaah dokumen. Metode-metode pengumpulan data tersebut meliputi :

1. Pengamatan *(Observasi)*

   Yaitu dengan cara mengamati secara langsung dengan cara mempelajari secara langsung aksara lontara.

2. Telaah Dokumen

   Yaitu suatu proses pengumpulan dokumen-dokumen yang berkaitan dengan penelitian.

D. Alat dan Bahan Penelitian

Dalam pelaksanaan peneltian ini digunakan beberapa alat dan bahan yang terdiri dari Perangkat Keras (Hardware) dan Perangkat Lunak (Software). Perangkat keras yang digunakan terdiri dari Laptop *ASUS* dengan prosesor *AMD Ryzen 3 3250U*, kapasitas *RAM 4GB* dan Smartphone Android. Perangkat lunak yang digunakan adalah Windows 10 Home *Single Language*, *Android Studio*, *Photoshop* Sedangkan Sumber data pada penelitian ini diperoleh dari observasi dan telaah di lokasi penelitian.

E. Tahap Penelitian

Tahap yang dilakukan penulis dalam perancangan aplikasi ini adalah sebagai berikut :

1. Pengumpulan data

   Mengumpulkan data - data dan informasi untuk di jadikan acuan dalam membangun aplikasi yang dirancang.

2. Analisis sistem

   Mengidentifikasi dan mengenali masalah yang ada, kemudian mencari alternative - alternative pemecahannya.

3. Desain sistem

   Setelah masalah ditentukan dan analisa data sudah dilakukan maka perlu di lakukan pembangunan atau mendesain sistem tentang masalah yang sudah ditentukan.

4. *Coding*

   Tahap penulisan kode *(listing)* program sistem.

5. Pengujian sistem

   Setelah proses *coding* selesai, dilakukan pengujian sistem menggunakan metode *black box.*

6. Implementasi sistem

   Tahap ini di gunakan untuk penerapan dan pengujian sistem ke dalam kondisi sebenarnya agar dapat diketahui kekurangan dan kelebihannya.

7. Penulisan Laporan

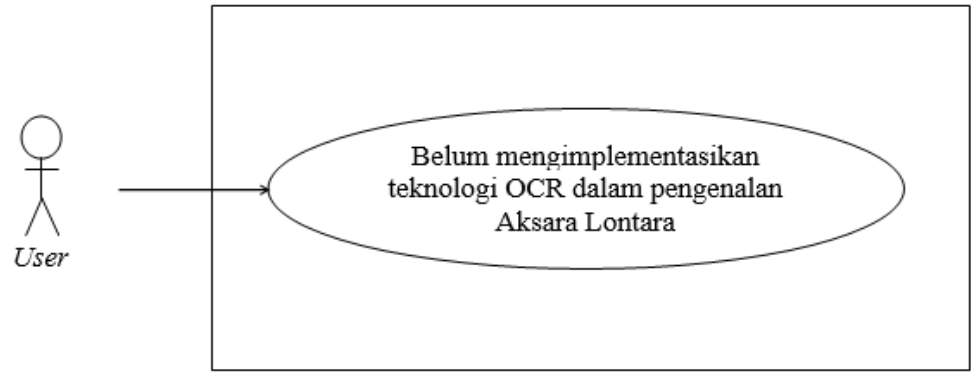
   Pada tahap ini di gunakan untuk menulis laporan-laporan.

F. Metode Pengujian

Pengujian dilakukan dengan menguji setiap proses yang ada pada sistem. Pengujian ini dilakukan secara *WhiteBox*, yaitu pengujian terhadap cara kerja perangkat lunak itu sendiri yaitu prosedur programnya (*basis path*) atau proses *looping* (pengulangan). Dan juga *BlackBox*, yaitu pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak.

G. Analisis Sistem

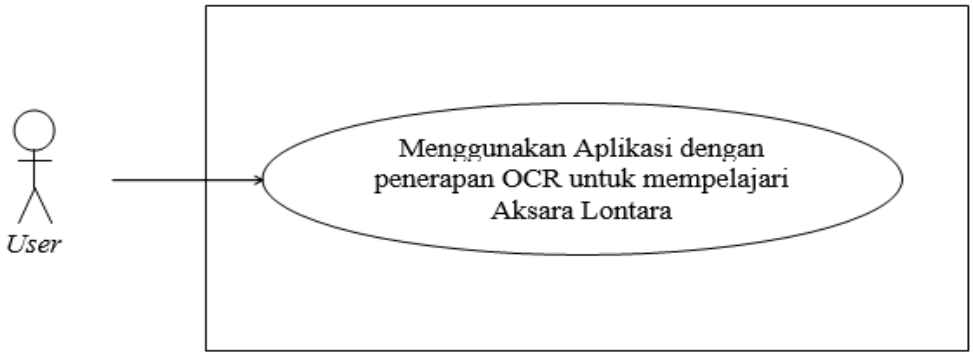
1. Sistem Yang Berjalan



Gambar 3. 1. Sistem Yang Berjalan

Sistem yang berjalan, belum adanya aplikasi yang mengimplementasikan teknologi OCR dalam menerjemahkan Aksara Lontara sehingga masyarakat tidak mengenal Aksara Lontara itu sendiri.

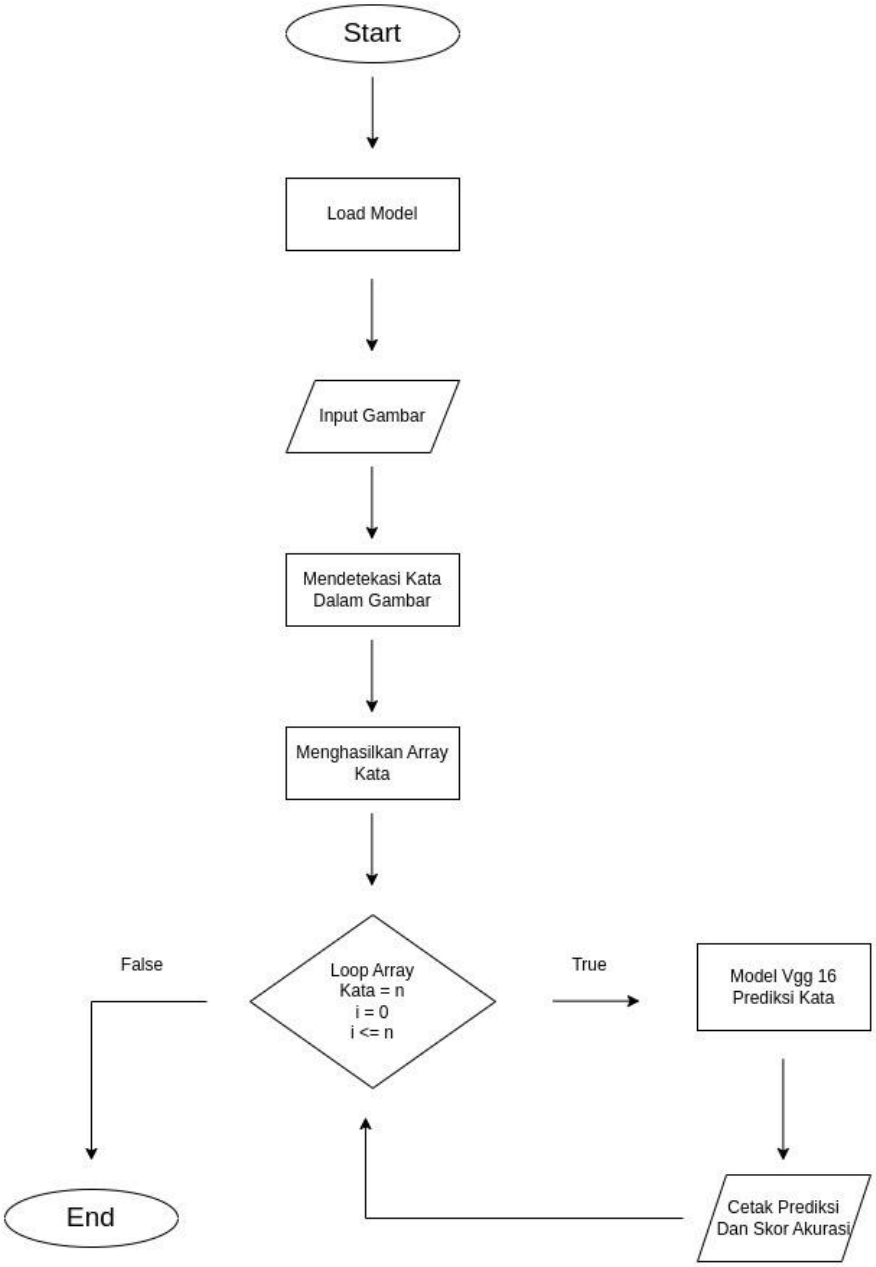
2. Sistem Yang Diusulkan



Gambar 3. 2. Sistem Yang Diusulkan

Sistem yang diusulkan, dengan membuat aplikasi dengan menerapkan teknologi OCR dapat membantu masyarakat dalam mempelajari Aksara Lontara yang berada disekitar mereka.

3. Flowchart Sistem



Gambar 3. 3. Flowchart Sistem

Pada penggunaan aplikasi dengan menginisialisasi untuk mengaktifkan kamera, kemudian menyorot image atau gambar Aksara

Lontara setelah itu program akan menjalankan proses teknologi OCR untuk menampilkan terjemahan Aksara Lontara.

BAB IV

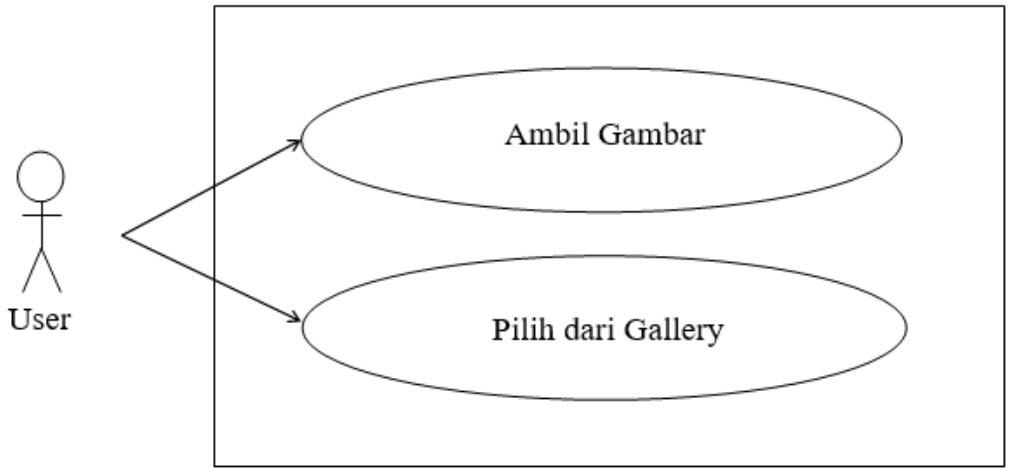
HASIL DAN PEMBAHASAN

A. Analisis Aliran Data Dengan UML

Analisis aliran data bertujuan mengetahui aliran proses informasi. Dalam analisis sistem ini, penulis menggunakan pengembangan orientasi objek sehingga menggunakan *Use Case* Diagram, *Activity* Diagram dan *Sequence* Diagram.

1. *Use Case* Diagram

Use Case Diagram berfungsi untuk menjalankan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*).



Gambar 4. 1 Use Case User

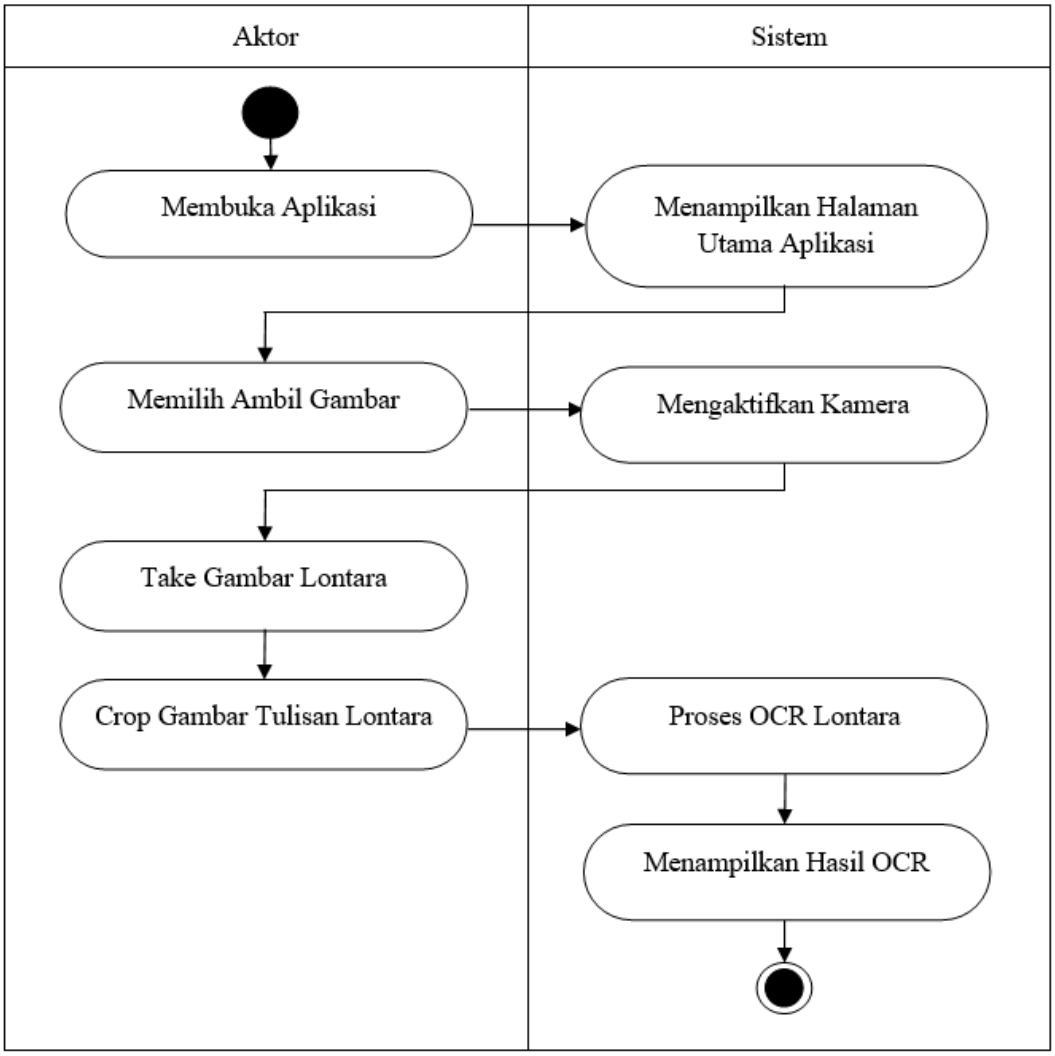
Tabel 4. 1 Penjelasan Use Case Diagram User

Nama *Use Case*	Deskripsi *Use Case*
Ambil Gambar	Proses dimana *User* mengambil gambar langsung melalui device kamera *handphone*.
Pilih dari Gallery	Proses dimana *User* memilih gambar dari *Gallery* handphone

2. *Activity Diagram*

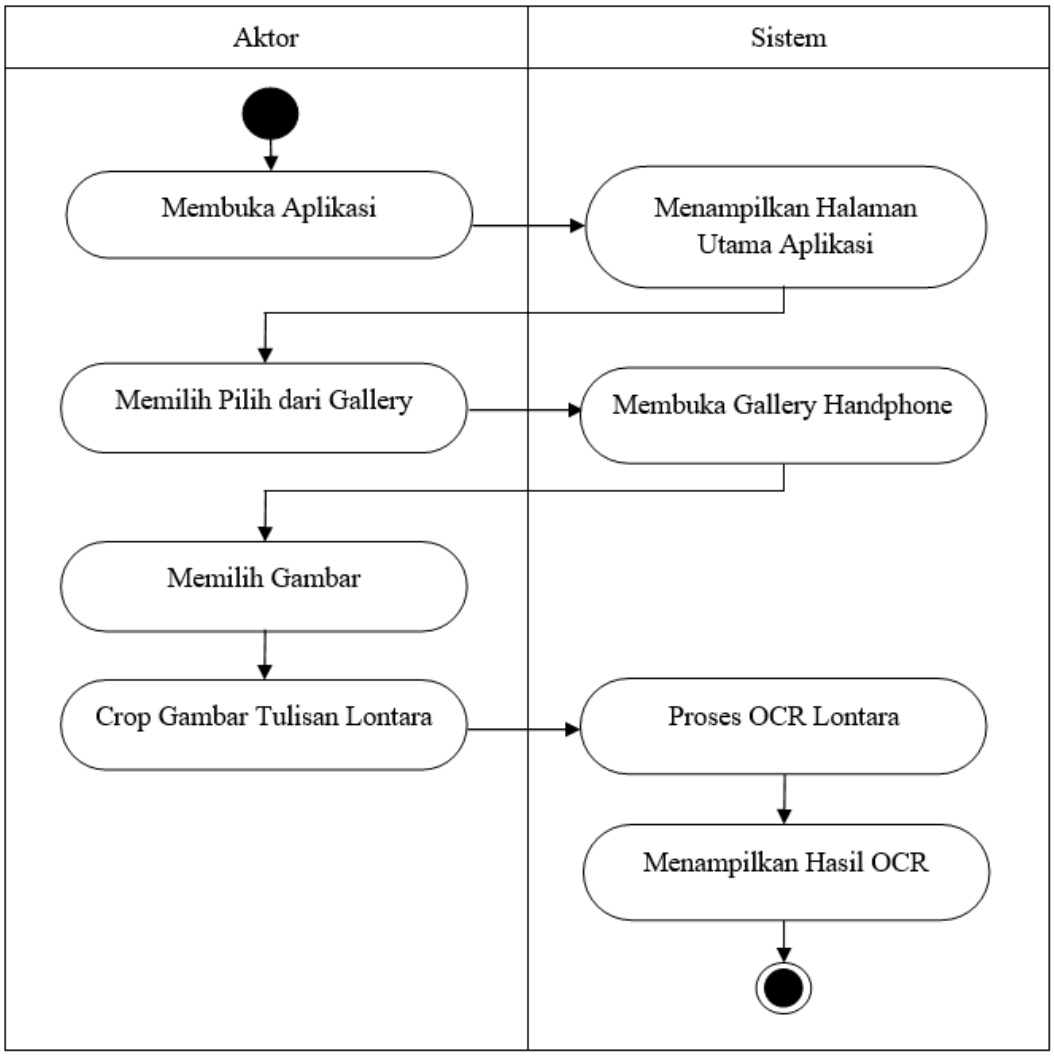
Activity Diagram ini menjelaskan tentang aktivitas - aktivitas yang terjadi dalam sebuah aliran proses pada sistem.

a. Diagram *Activity* Ambil Gambar



Gambar 4. 2 Activity Diagram Ambil Gambar

b. Diagram Activity Pilih dari Gallery

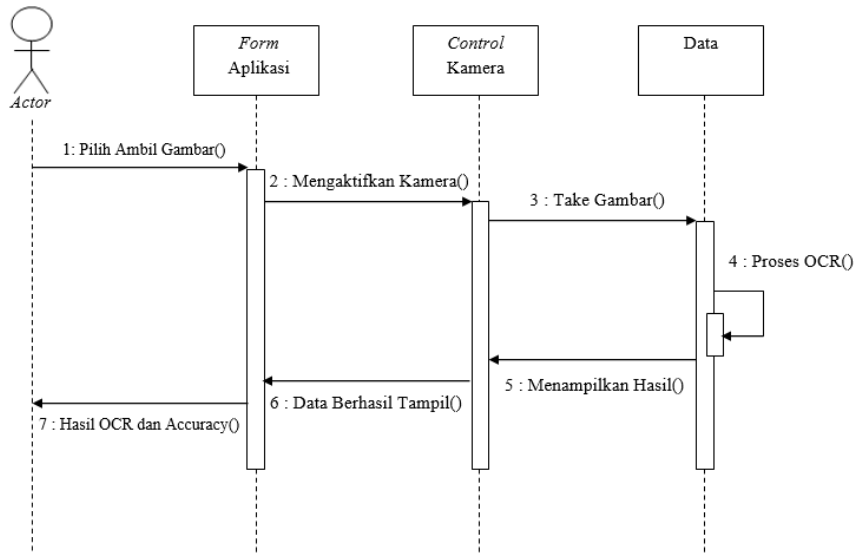


Gambar 4. 3 Activity Diagram Pilih dari Gallery

3. *Sequence* Diagram

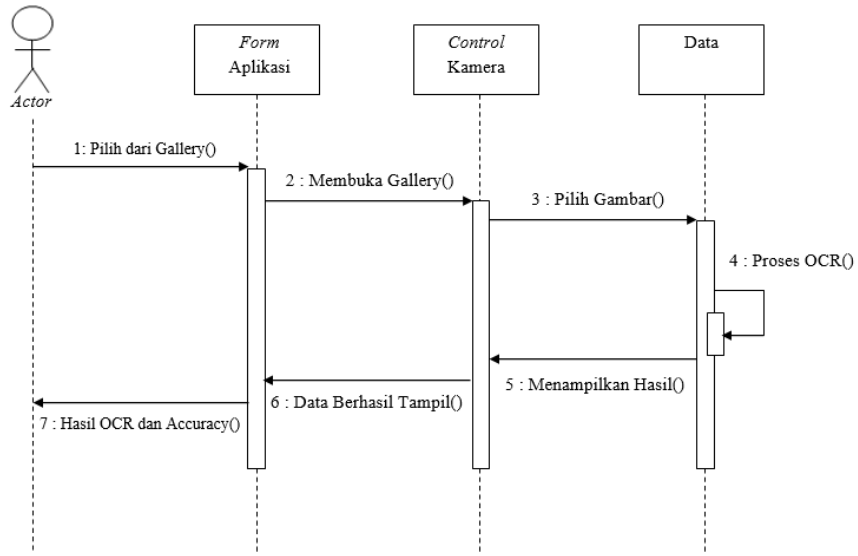
Sequence Diagram merupakan aliran antara objek yang membentuk proses, berikut adalah diagram *sequencenya*.

a. Diagram *Sequence* Ambil Gambar



Gambar 4. 4 Diagram *Sequence* Ambil Gambar

b. Diagram *Sequence* Pilih dari Gallery

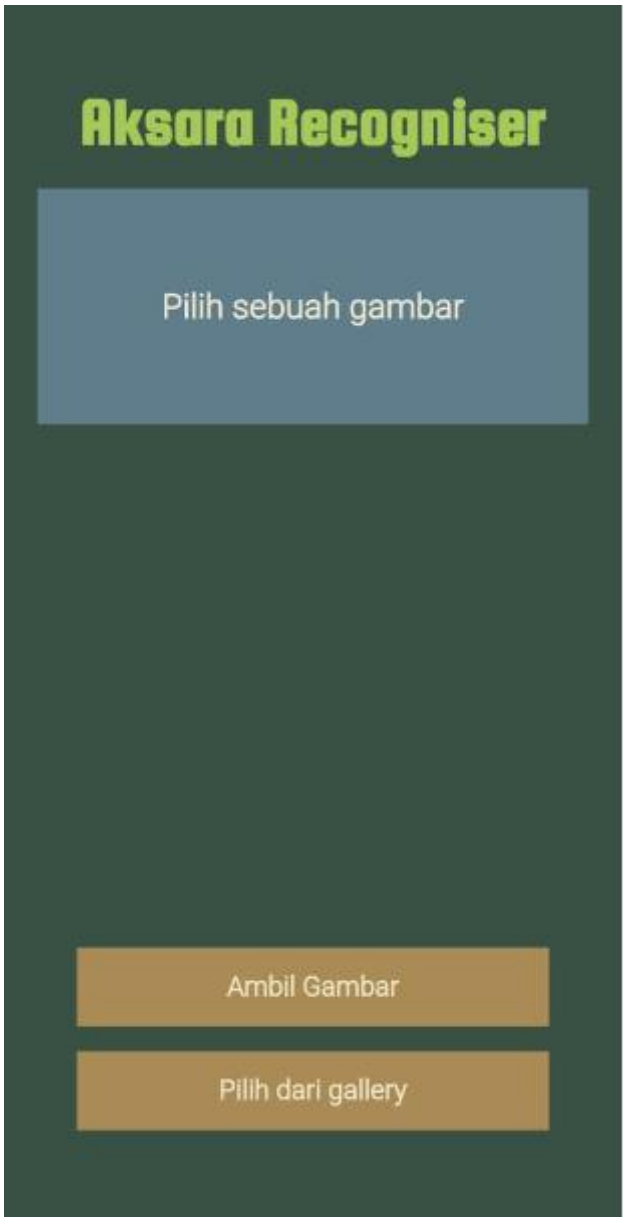


Gambar 4. 5 Diagram Sequence Pilih dari Gallery

B. Rancangan Aplikasi

1. *Form* Utama

Merupakan tampilan *Form* Utama yang digunakan *user* untuk mengelola aplikasi.



Gambar 4. 6 *Form* Utama

Ket script jika mengambil gambar:

class _MyAppState extends State<MyApp> {

```
final controller = UltralyticsYoloCameraController();

@override
Widget build(BuildContext context) {
return MaterialApp(
home: Scaffold(
body: FutureBuilder<bool>(
future: _checkPermissions(),
builder: (context, snapshot) {
final allPermissionsGranted = snapshot.data ?? false;

if (!allPermissionsGranted) {
return const Center(child: Text('Permissions not granted'));
}

return FutureBuilder<ObjectDetector>(
future: _initObjectDetectorWithLocalModel(),
builder: (context, snapshot) {
if (snapshot.connectionState == ConnectionState.waiting) {
return const Center(child: CircularProgressIndicator());
}

final predictor = snapshot.data;
```

```
if (predictor == null) {
return const Center(child: Text('Failed to initialize predictor'));
}

return Stack(
children: [
UltralyticsYoloCameraPreview(
controller: controller,
predictor: predictor,
onCameraCreated: () {
predictor.loadModel(useGpu: true);
},
),
StreamBuilder<double?>(
stream: predictor.inferenceTime,
builder: (context, snapshot) {
final inferenceTime = snapshot.data;

return StreamBuilder<double?>(
stream: predictor.fpsRate,
builder: (context, snapshot) {
final fpsRate = snapshot.data;
```

```
return Times(
inferenceTime: inferenceTime,
fpsRate: fpsRate,
);
},
);
},
),
],
);
},
);
},
),
floatingActionButton: FloatingActionButton(
child: const Icon(Icons.cameraswitch),
onPressed: () {
controller.toggleLensDirection();
},
),
),
);
```

}

2. Ambil Gambar

Merupakan tampilan Kamera yang digunakan *user* untuk mengambil gambar.



Gambar 4. 7 Ambil Gambar

3. Pilih dari Gallery

Merupakan tampilan *Gallery* yang digunakan *user* untuk memilih gambar.



Gambar 4. 8 Pilih dari Gallery

Ket script jika mengambil dari galery:

```
class Times extends StatelessWidget {
const Times({
super.key,
required this.inferenceTime,
required this.fpsRate,
});

final double? inferenceTime;
final double? fpsRate;

@override
Widget build(BuildContext context) {
return SafeArea(
child: Align(
alignment: Alignment.bottomCenter,
child: Container(
margin: const EdgeInsets.all(20),
padding: const EdgeInsets.all(20),
decoration: const BoxDecoration(
borderRadius: BorderRadius.all(Radius.circular(10)),
color: Colors.black54,
),
```

```
child: Text(
'${(inferenceTime ?? 0).toStringAsFixed(1)} ms  -  ${(fpsRate ??
0).toStringAsFixed(1)} FPS',
style: const TextStyle(color: Colors.white70),
)),
),
);
}
}
```

C. Implementasi

Implementasi sistem merupakan tahap penerapan dari suatu teknologi yang didesain untuk siap dioperasikan. Tahap ini merupakan terjemahan perancangan dari bab hasil analisis sebelumnya dalam suatu bahasa pemrograman. Bahasa pemrograman yang digunakan untuk membangun aplikasi menggunakan *Android Studio* adalah bahasa pemrograman *Kotlin.*

1. Kebutuhan perangkat keras

   Spesifikasi minimum perangkat keras sebagai berikut :

   Tabel 4. 2 Kebutuhan Perangkat Keras

Jenis	**Spesifikasi**
Laptop	*ASUS WIN-B0GMRJJET7E*
Processor	*Intel(R) Celeron(R) CPU 847 @ 1.10GHz 1.10 GHz*
Memory	4GB RAM

Harddisk	500GB

2. Kebutuhan perangkat lunak

Spesifikasi minimum perangkat lunak sebagai berikut :

Tabel 4. 3 Kebutuhan Perangkat Lunak

Jenis	Spesifikasi
Sistem Operasi	*Windows 10 PRO - 64 Bit*
Alat Pengembangan	*Android Studio*

D. Pengujian Sistem

1. *BlackBox*

Pengujian sistem dilakukan dengan cara pengujian *BlackBox.*

Tabel 4. 4 *BlackBox Form* Utama

Test Faktor	Hasil	Kesimpulan
Jika *user* membuka aplikasi *aksara recogniser*.	✓	Informasi, tampil *form* utama.
Screen Shot		



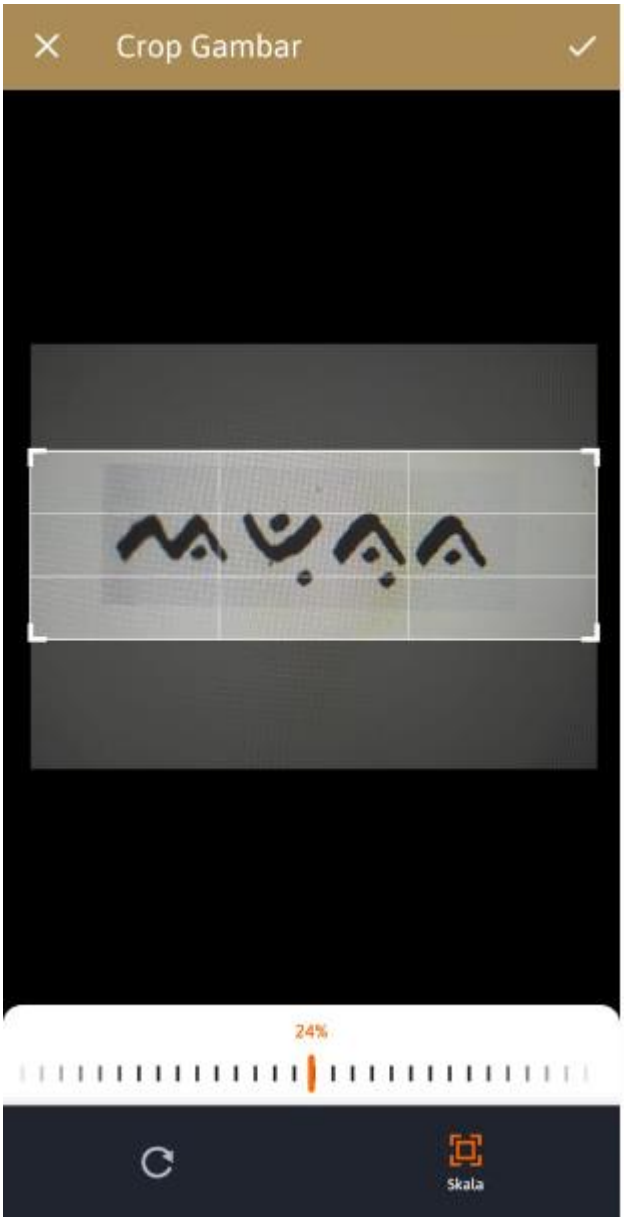
Tabel 4. 5 *BlackBox* Kamera Device

Test Faktor	Hasil	Kesimpulan
Jika *user* menekan menu ambil gambar.	✓	Informasi, tampil kamera *device.*
Screen Shot		



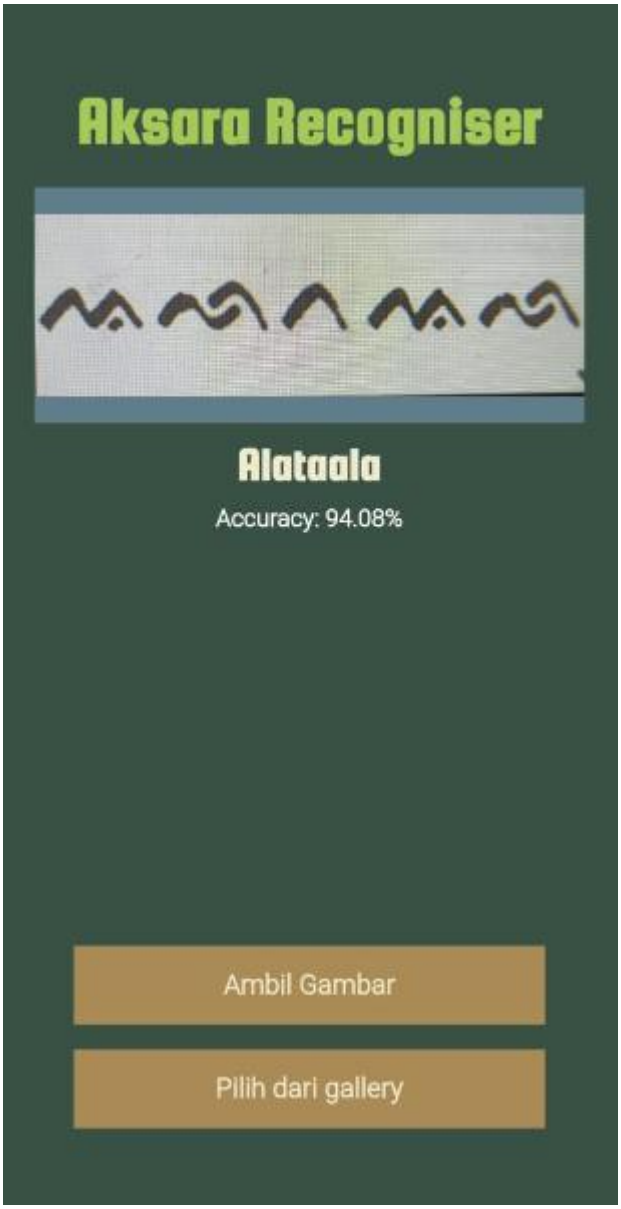
Tabel 4. 6 *BlackBox* Crop Gambar

Test Faktor	Hasil	Kesimpulan
Jika *user* telah menggambil gambar lontara.	✓	Informasi, tampil *crop* gambar.
Screen Shot		



Tabel 4. 7 *BlackBox* Hasil OCR

Test Faktor	Hasil	Kesimpulan
Jika *user* telah menyesuaikan *crop* gambar dan menekan tombol ceklis.	✓	Informasi, tampil hasil OCR.
Screen Shot		



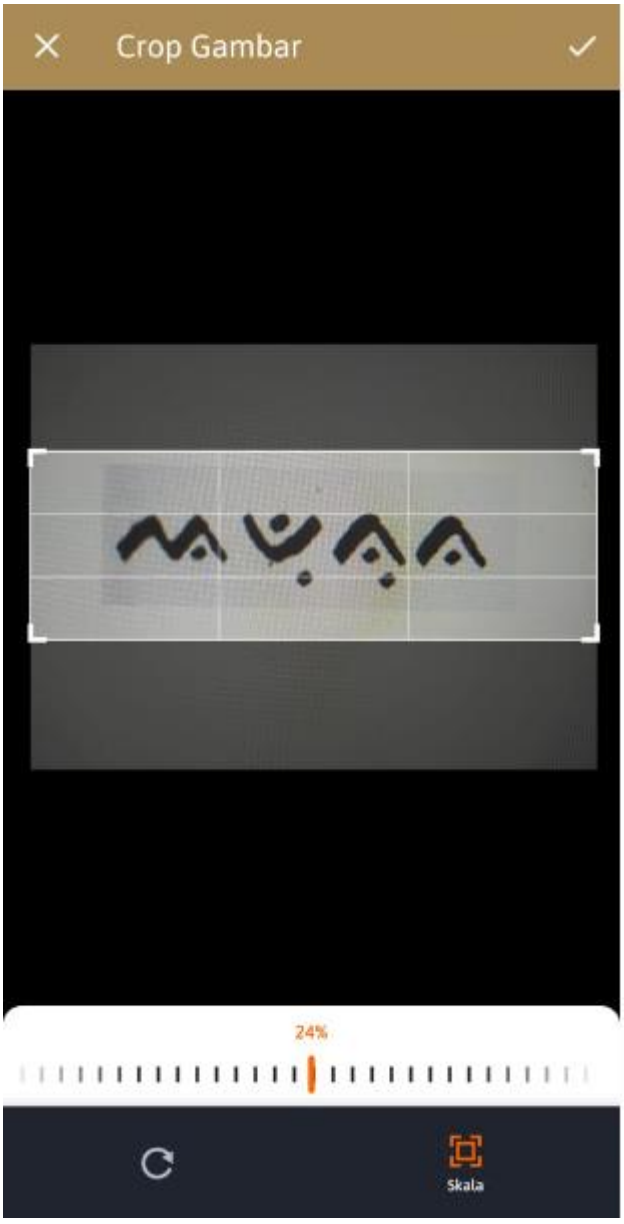
Tabel 4. 8 *BlackBox* Gallery Handphone

Test Faktor	Hasil	Kesimpulan
Jika *user* menekan menu pilih dari *gallery*.	✓	Informasi, tampil *gallery* handphoe.

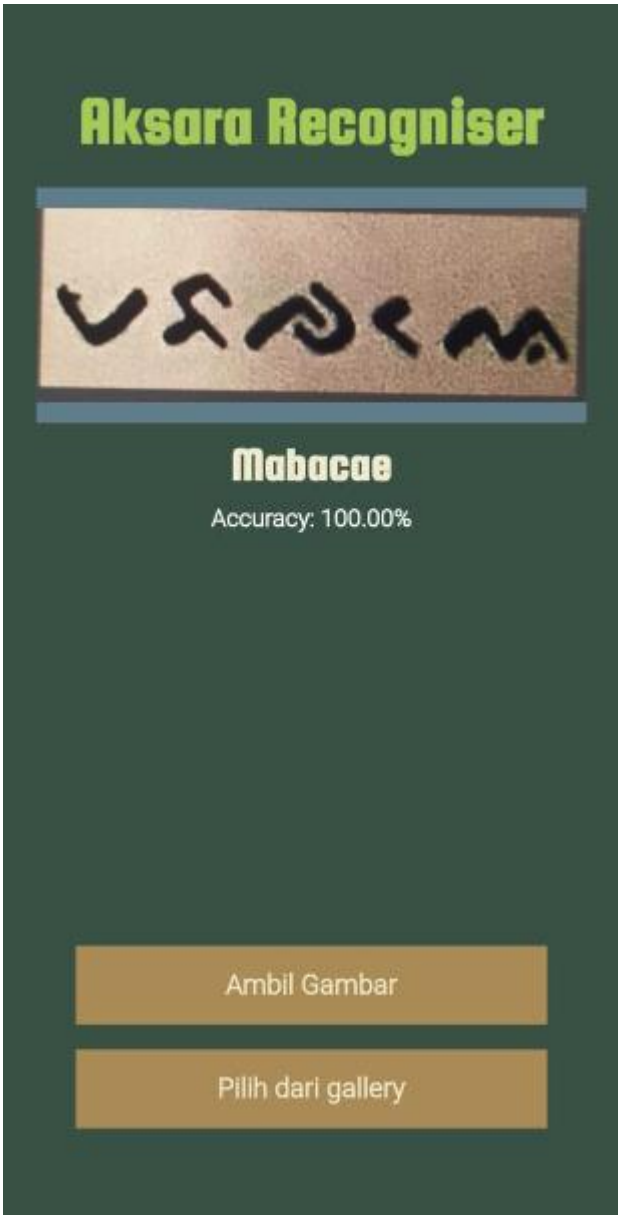
Screen Shot


Tabel 4. 9 *BlackBox* Crop Gambar

Test Faktor	Hasil	Kesimpulan

Jika *user* memilih salah satu gambar pada *gallery*.	✓	Informasi, tampil crop gambar.
Screen Shot		
		

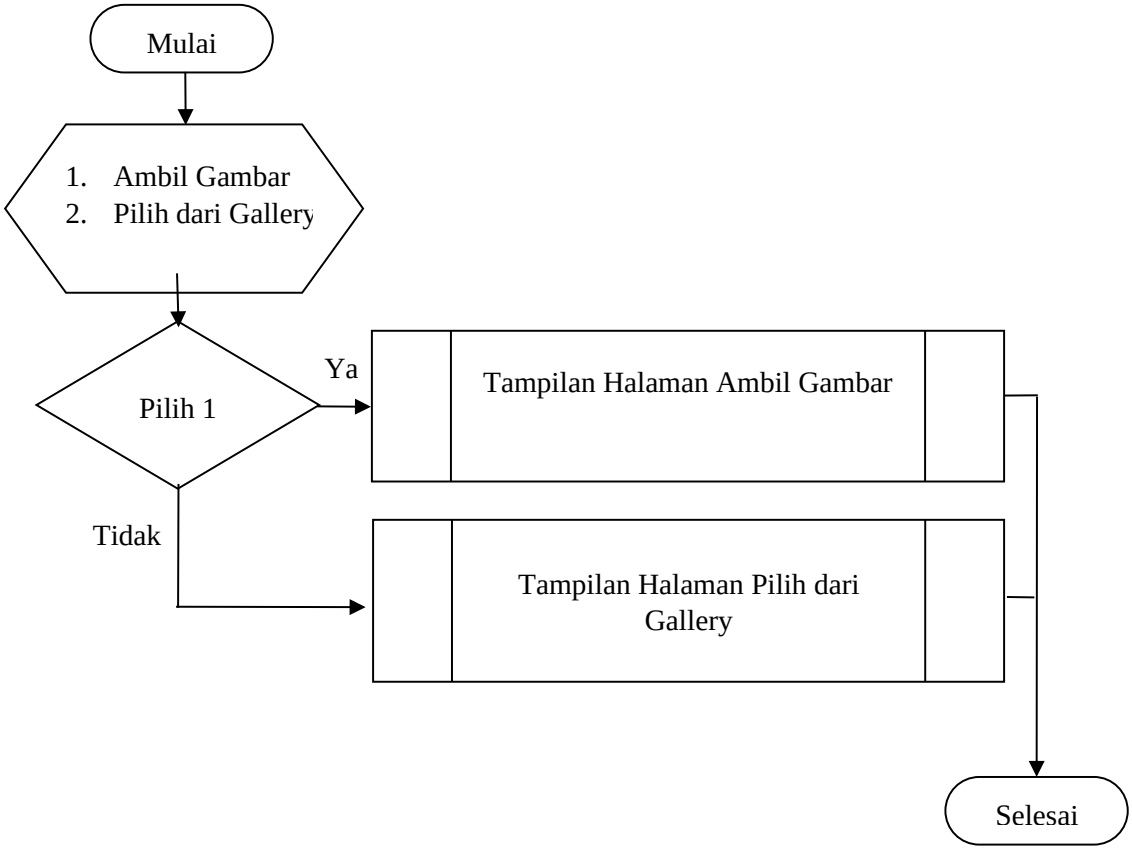
Tabel 4. 10 *BlackBox* Hasil OCR

Test Faktor	Hasil	Kesimpulan
Jika *user* telah menyesuaikan crop gambar dan menekan tombol ceklis.	✓	Informasi, tampil hasil OCR.
Screen Shot		
		

2. *WhiteBox*

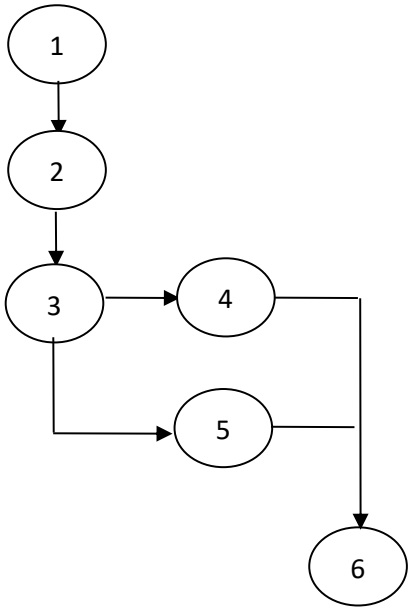
Pengujian aplikasi dilakukan dengan cara pengujian *White Box:*

a. *Flowchart* dan *Flowgraph* Aktivitas *User*



Gambar 4. 9. *Flowchart* Aktivitas *User*

Dari *Flowchart* yang digunakan untuk pengujian perangkat lunak, maka ditentukan *Flowgraph* sebagai berikut:



Gambar 4. 10. *Flowgraph* Aktivitas *User*

Dari *Flowgraph* aktivitas *User* diatas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node*:

   Dengan rumus : V(G) = E – N + 2

   E (*edge*) = 6

   N (*Node*) = 6

   P (Predikat *Node*) = 3

   Penyelesaian : V(G) = E – N + 2

   = 6 – 6 + 2

   = 2

Predikat (P) = P + 1

= 1 + 1

= 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2
3. *Independent path* pada *Flowgraph* diatas adalah:

   Path 1 = 1 – 2 – 3 – 4 – 6

   Path 2 = 1 – 2 – 3 – 5 – 6
4. Grafik Matriks Aktivitas *User*

Tabel 4. 11. Grafik Matriks Aktivitas *User*

	1	2	3	4	5	6	E – 1
1		1					1 – 1 = 0
2			1				1 – 1 = 0
3				1	1		2 – 1 = 1
4						1	1 – 1 = 0
5						1	1 – 1 = 0
6							0
SUM (E+1)							1 + 1 = 2

b. Hasil Pengujian

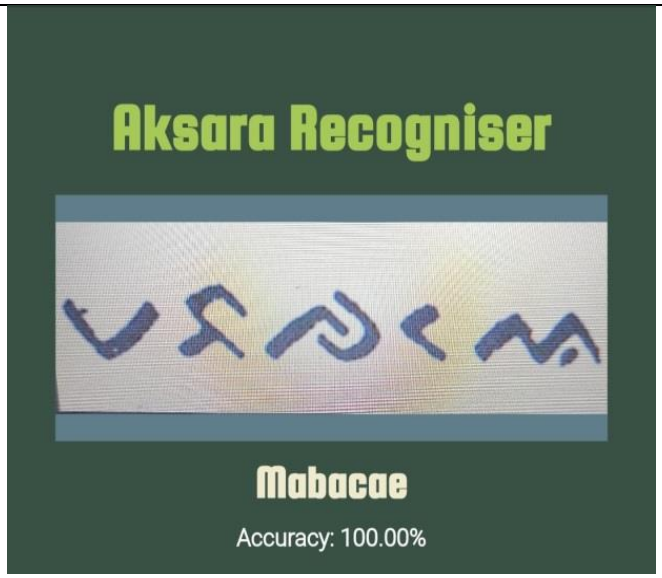
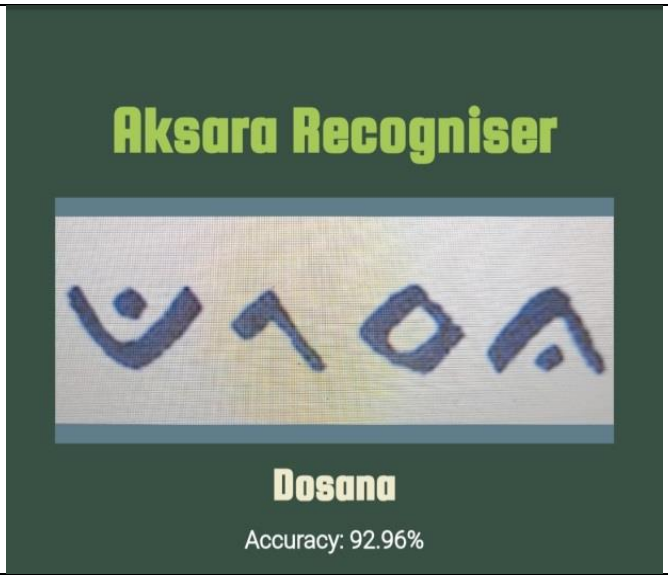
Tabel 4. 12. Hasil Pengujian

No	***Flowchart***	***Independent Path***	***Region***	***Cyclomatic Complexity***	
1	Aktivitas *User*	2	2	2	

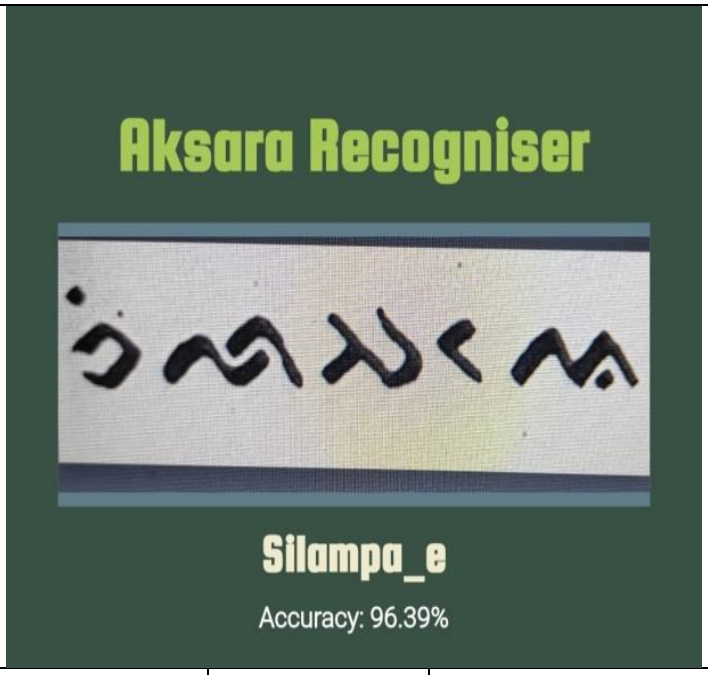
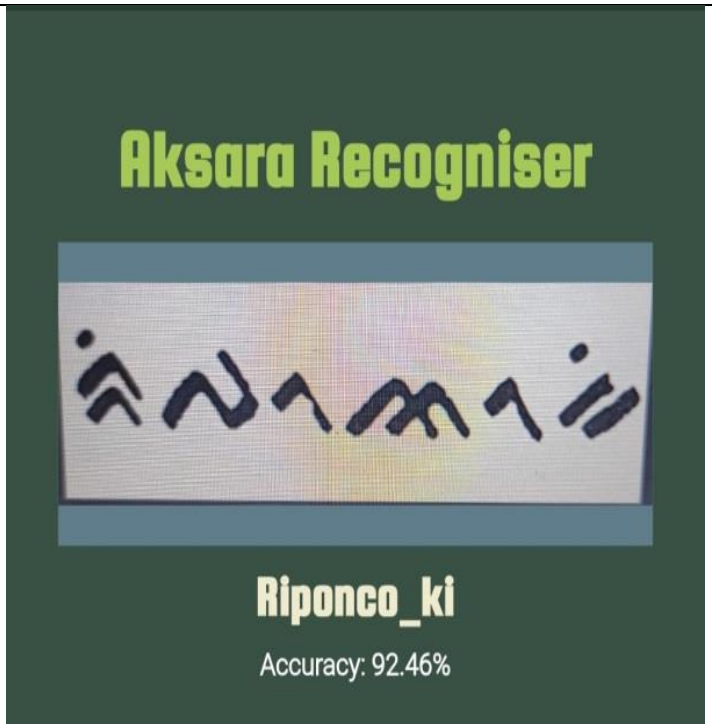
3. Pengujian Tingkat Akurasi

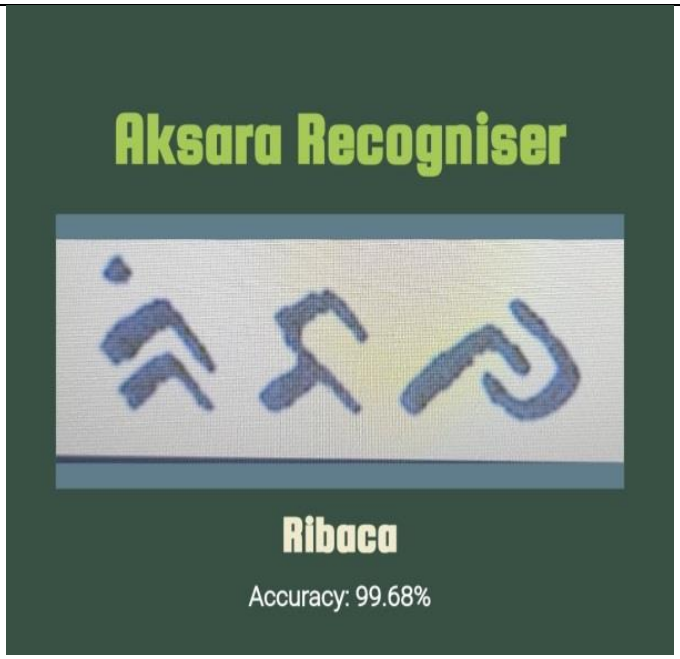
Pengujian aplikasi untuk mendapatkan tingkat akurasi deteksi.

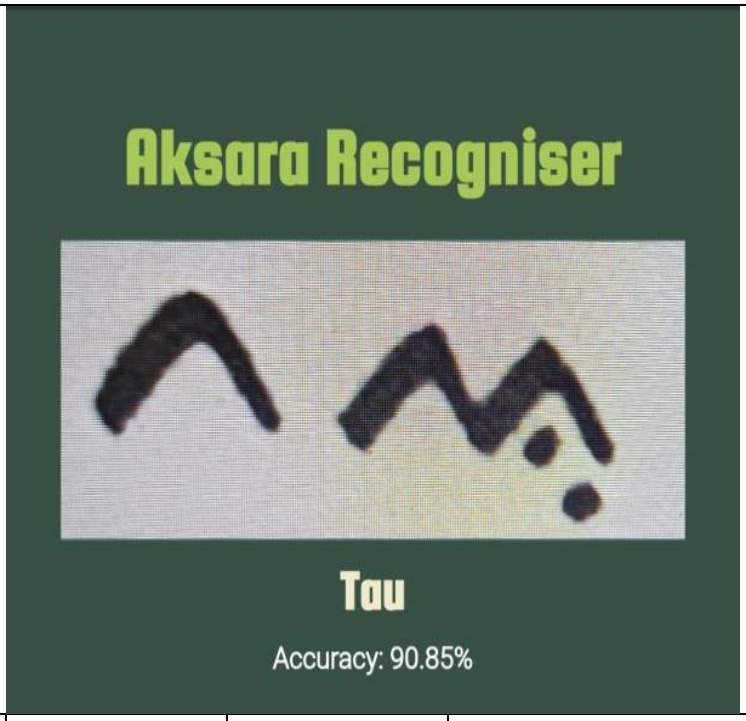
Tabel 4. 13. Hasil Pengujian Tingkat Akurasi

<table>
<tr><td>No</td><td>Aksara Lontara</td><td>Artinya</td><td>Tingkat Akurasinya (%)</td></tr>
<tr><td>1</td><td>Mabacae</td><td>Membaca</td><td>100%</td></tr>
<tr><td colspan="4">Screenshot</td></tr>
<tr><td colspan="4">
</td></tr>
<tr><td>2</td><td>Dosana</td><td>Dosanya</td><td>92,96%</td></tr>
<tr><td colspan="4">Screenshot</td></tr>
<tr><td colspan="4">
</td></tr>
</table>

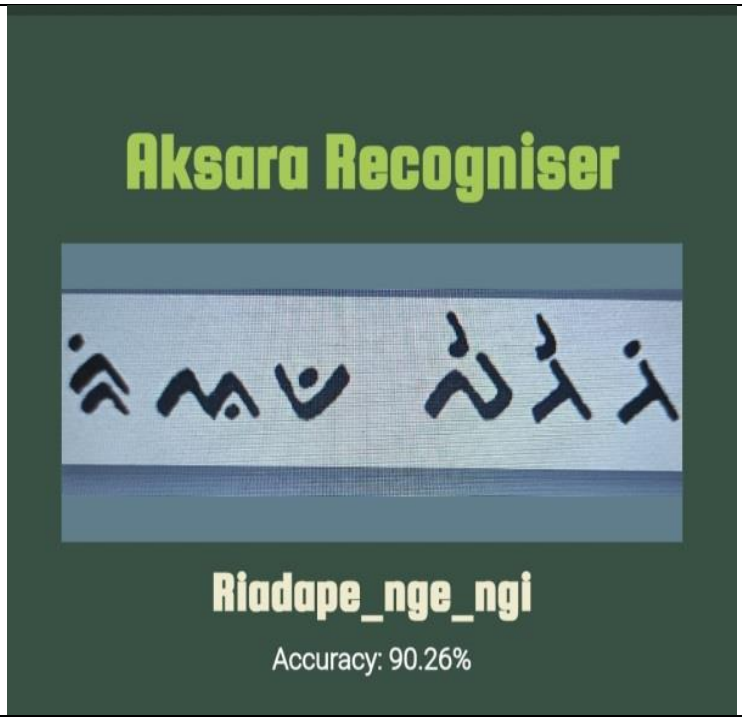
No	Aksara Lontara	Artinya	Tingkat Akurasinya (%)
3	Deto	Tidak	99,98%
Screenshoot			
			
4	Alataala	Allah SWT	98,70%
Screenshot			
			

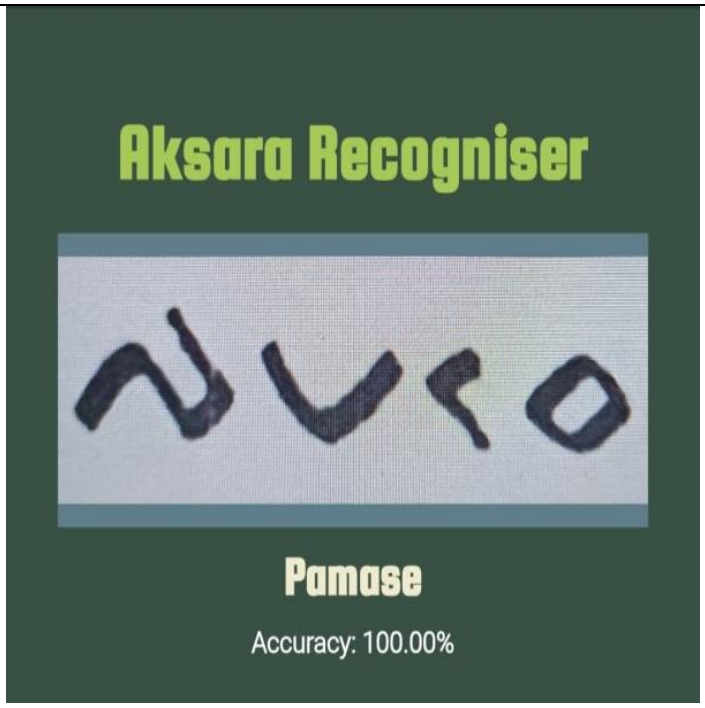
No	Aksara Lontara	Artinya	Tingkat Akurasinya (%)
5	Silampae	Kejar-kejaran	96,39%
Screenshot			
			
6	Riponco ki	Pendek	92,46%
Screenshot			
			

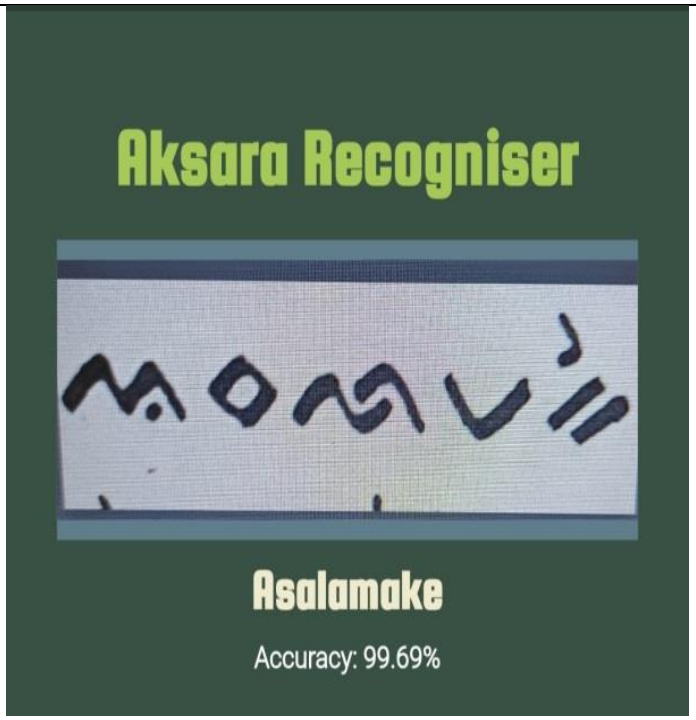
No	Aksara Lontara	Artinya	Tingkat Akurasinya (%)
7	Ricapana	Ujungnya	99,62%
Screenshot			
			
8	Ribaca	Dibaca	99,68%
Screenshot			
			

No	Aksara Lontara	Artinya	Tingkat Akurasinya (%)
9	Tau	Orang	90,85%
Screenshot			
			
10	Naonro	Ditempati	96,18%
Screenshot			
			

No	Aksara Lontara	Artinya	Tingkat Akurasinya (%)
11	Naenkalinga	Mendengarkan	99,91%
Screenshot			
			
12	Taroengi	Disimpan	96,94%
Screenshot			
			

No	Aksara Lontara	Artinya	Tingkat Akurasinya (%)
13	Ripamulana	Dimulainya	100%
Screenshot			
			
14	Riadapengengi	Dimaafkan	90,26%
Screenshot			
			

No	Aksara Lontara	Artinya	Tingkat Akurasinya (%)
15	Sureng	Teman	99,45%
Screenshot			
			
16	Pamase	Kasihan	100%
Screenshot			
			

No	Aksara Lontara	Artinya	Tingkat Akurasinya (%)
17	Made ne e	Terjatuh	97,50%
Screenshot			
			
18	Asalamake	Keselamatan	99,69%
Screenshot			
			

No	Aksara Lontara	Artinya	Tingkat Akurasinya (%)
19	Aregi	Ataukah	96,35%
Screenshot			
			

Penggunaan aplikasi sangat mudah dengan memilih gambar dari gallery ataupun mengambil gambar langsung kemudian tentukan gambar huruf Aksara Lontara yang ingin di terjemanhkan dengan cara mengcrop gambar kemudian aplikasi akan automatis mendeteksi menggunakan teknologi OCR yang telah diterapkan kemudian hasil terjemahan akan tampil.

BAB V

PENUTUP

A. Kesimpulan

Setelah dilakukanya pengujian pada aplikasi Aksara *Recogniser*, maka penulis mendapatkan beberapa kesimpulan sebagai berikut :

1. Aplikasi optical character recognition penegenalan tulisan aksara lontara berbasis android. Menggunakan Phyton sebagai machine learning dan flutter sebagai frontend (Android) untuk menangkap gambar dan memprosesnya dengan metode OCR (*Optical Character Recognition*).
2. Dataset yang digunakan dalam penegenalan tulisan aksara yaitu: 𑻠 , 𑻡, 𑻢, 𑻣, 𑻤, 𑻥, 𑻦,𑻧,𑻨,𑻩,𑻪,𑻫,𑻬,𑻭,𑻮,𑻯,𑻰,𑻱,𑻲,𑻳,𑻴,𑻵,𑻶,𑻷,𑻸
3. Proses pengenalan aksara dilakukan dengan scanning gambar menggunakan android lalu diproses melalui Phyton dengan metode OCR (*Optical Character Recognition*).
4. Adapun metode /Algoritma yang digunakan adalah menggunakan Backbone Network yaitu Model CNN yang digunakan sebagai backbone untuk mengekstrak fitur, kemudian mendeteksi objek dengan ukuran yang berbeda (misalnya area aksara lontara).
5. Tingkat dalam mengenali pola yang benar dengan Tingkat pembacaan OCR adalah rata – rata 96,73%, dengan perhitungan jika akurasi gambar yang di kenali di atas 0.5 sampai 1, maka pembacaan aksara memiliki akurat pembacaan 70% - 96,73% %

B. Saran

Saran yang penulis berikan untuk arah perkembangan selanjutnya agar perancangan aplikasi ini dapat berkembang sesuai dengan teknologi yaitu, dengan menambahkan atau memperbanyak detaksi aksara lainnya.

DAFTAR PUSTAKA

A) Kuswanar, R. 2019. Proses OCR dengan Onipage PRO 10. Yogyakarta: Elex Media.

B) Wahid, S. dkk. 2020. Pattimpa Bajae Metode TST (Tulisan, Sebutan, Terjemahan). Makassar: Badan Penerbit UNM.

C) Kadir, A. 2020. Pemrograman Android & Database. Yogyakarta: Andi Offsite.

D) Pamungkas. 2017. Pengelolaan Citra Digital. Jakarta: Erlangga.

E) Herlinah & Musliadi. 2021. Pemrograman Aplikasi Android Dengan Android Studio Photoshop dan Audition. Yogyakarta: Andi Offsite.

F) Arifianto, R. 2017. *Materi Pembahasan Flowchart. From* https://rahmatarifianto. wordpress.com/2014/11/20/pengertian-*Flowchart*-dan-jenis-jenisnya.html. (20 November 2023).

G) Sora, N. 2020. *Pengertian UML dan Jenis - jenisnya*. *From* http://www. pengertianku.net/2015/09/pengertian-uml-dan-jenis jenisnya-serta-contoh-diagramnya.html. (20 November 2022).

H) Suryo Hartanto, dkk. 2018. "Optical Character Recognition Menggunakan Algoritma Template Matching Correlation," Jurnal Masyarakat Informatika, vol. 5, no. 9, pp. 1-12, Apr. 2015. https://doi.org/10.14710/jmasif.5.9.8435.

I) Fredian, O. 2016. “Aplikasi Pengenalan Plat Nomor Kendaraan Dengan Metode Optical Character Recognition (OCR)”. Skripsi: Universitas Muhammadiyah Surakarta.

J) Andono, Pulung N; Sutojo, T.; , Muljono;, in PENGOLAHAN CITRA DIGITAL, Yogyakarta, ANDI, 2017, p.

K) F. Liantoni, "Pengenalan karakter angka menggunakan metode Integral Proyeksi," Jurnal Ilmiah Teknologi Sistem Informasi, pp. 57-64, 2018.

L) P. Pangestu, "Penerapan Histogram Equalization pada Optical Character Recognition Preprocessing," Ultimatics: Jurnal Teknik Informatika, vol. VII, no. 1, pp. 27-34, 2015.