



## INTEGRASI MODEL *DEEP LEARNING* PADA PENGEMBANGAN APLIKASI ANDROID PENDETEKSI BAHASA ISYARAT SIBI MENGGUNAKAN *JETPACK COMPOSE*

Raka Agus Maulana<sup>1</sup>, Ahmad Rio Adriansyah<sup>2</sup>, Zaki Imaduddin<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
Jakarta Selatan, DKI Jakarta, Indonesia 12640

rakh21186ti@student.nurulfikri.ac.id, arasy@nurulfikri.ac.id, zaki@nurulfikri.ac.id

### Abstract

Communication barriers between the deaf community and the general public persist due to a limited understanding of sign language. This study aims to develop an Android application capable of recognizing static alphabet letters in the Indonesian Sign Language (SIBI) in real time to support inclusive communication. The application was developed using the Kotlin programming language and Jetpack Compose, and follows the Clean Architecture approach to ensure modularity and ease of maintenance. A machine learning model was integrated using TensorFlow Lite to enable optimal performance on mobile devices. The development process adopted the Rapid Application Development (RAD) methodology, including stages of literature review, system design, implementation, and testing. Black Box testing showed that all core features functioned as expected. Furthermore, user acceptance testing (UAT) showed a satisfaction rate of 91.3%, indicating a positive perception of usability and interface design. This application is expected to serve as an initial solution to help reduce communication gaps between the deaf community and the broader society.

**Keywords:** Android, Deep Learning, Rapid Application Development, SIBI, TensorFlow Lite

### Abstrak

Hambatan komunikasi antara penyandang tunarungu dan masyarakat umum masih menjadi persoalan akibat kurangnya pemahaman terhadap bahasa isyarat. Penelitian ini bertujuan untuk mengembangkan aplikasi Android yang mampu mengenali huruf alfabet statis dalam Sistem Isyarat Bahasa Indonesia (SIBI) secara *real-time* guna mendukung komunikasi yang inklusif. Aplikasi ini dikembangkan menggunakan bahasa pemrograman Kotlin dan Jetpack Compose, serta menerapkan arsitektur perangkat lunak Clean Architecture untuk meningkatkan modularitas dan kemudahan pemeliharaan. Model Deep Learning diterapkan melalui integrasi TensorFlow Lite agar dapat dijalankan secara optimal pada perangkat mobile. Proses pengembangan mengikuti metode Rapid Application Development (RAD) dengan tahapan studi literatur, desain, implementasi, dan pengujian. Hasil pengujian menggunakan metode Black Box menunjukkan bahwa semua fitur utama berfungsi sesuai dengan kebutuhan. Selain itu, uji penerimaan pengguna (UAT) menghasilkan tingkat kepuasan sebesar 91,3%, yang menandakan bahwa aplikasi diterima dengan baik dari segi kegunaan dan antarmuka. Aplikasi ini diharapkan dapat menjadi solusi awal dalam memperkecil kesenjangan komunikasi antara penyandang tunarungu dan masyarakat luas.

**Kata kunci:** Android, Deep Learning, Rapid Application Development, SIBI, TensorFlow Lite

### 1. PENDAHULUAN

Komunikasi adalah sebuah kegiatan umum manusia untuk menyampaikan informasi, namun terdapat orang yang memiliki masalah pendengaran dan berbicara, keterbatasan tersebut membuat mereka mengandalkan bahasa isyarat untuk berkomunikasi. Diindonesia sendiri terdapat 2 bahasa isyarat yang umum digunakan yaitu Sistem Bahasa Isyarat (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). SIBI adalah bahasa isyarat formal yang disusun dan diresmikan

oleh pemerintah Indonesia untuk digunakan pada sekolah luar biasa (SLB) [1], SIBI dalam pengembangannya mengadaptasi American Sign Language (ASL), yang hanya menggunakan gerakan tangan tunggal[2]. sedangkan BISINDO sendiri merupakan bahasa yang berkembang secara natural dalam komunitas penyandang distabilitas, serta memiliki karakteristik menyerupai bahasa asli. Menurut data WHO memperkirakan jumlah orang yang

mengalami pendengaran berkisar 466 juta orang, termasuk 34 juta adalah anak-anak atau 5% dari populasi dunia. [3].

Disisi lain, teknologi berkembang pesat, perangkat *smartphone* seperti android menawarkan potensi besar dalam membantu mengatasi hambatan ini. Pada data dari *GoodStats* menunjukkan pada tahun 2023 terdapat 209,3 juta pengguna aktif *smartphone* di Indonesia [4]. Memungkinkan penerapan model *machine learning* untuk mendeteksi bahasa isyarat SIBI dengan bantuan pustaka TensorFlow Lite dari Google. Iterasi ini memungkinkan aplikasi dapat mendeteksi bahasa isyarat secara *realtime*.

Penelitian sebelumnya yang dilakukan oleh Fikri Pratama Al Fajri [5] mengembangkan *model deep learning* untuk menerjemahkan bahasa isyarat SIBI dengan menggunakan metode *transfer learning* pada arsitektur MobileNetV2. Model tersebut dibangun dengan algoritma *Convolutional Neural Network* (CNN) dan menggunakan *dataset* gambar gerakan tangan dari platform Kaggle, yang terdiri dari huruf alfabet SIBI statis sebanyak 220 gambar per huruf, kecuali huruf Z dan J yang memerlukan gerakan dinamis. Hasil penelitian menunjukkan bahwa model mencapai tingkat akurasi sebesar 95,45% pada data pelatihan. Namun, penelitian tersebut hanya berfokus pada pembuatan dan evaluasi model di lingkungan pengembangan tanpa mengintegrasikannya ke dalam aplikasi *mobile* dan tanpa pengujian lebih lanjut di perangkat pengguna.

Metode *Rapid Application Development* (RAD) dipilih karena menekankan pengembangan cepat dan iterasi prototipe. Pendekatan ini memungkinkan pengguna terlibat langsung dalam pengujian dan validasi, sehingga menghasilkan aplikasi yang sesuai kebutuhan. Penelitian ini bertujuan membangun aplikasi penerjemah bahasa isyarat SIBI berbasis Android dengan Kotlin dan Jetpack Compose, mengintegrasikan *model deep learning*, serta menguji fungsionalitas dan penerimaan pengguna.

Penelitian ini diharapkan dapat menambah wawasan mengenai proses pengembangan aplikasi berbasis Android yang mampu mendeteksi bahasa isyarat SIBI secara *real-time*. Aplikasi ini bertujuan mempermudah komunikasi antara masyarakat umum dengan penyandang tunarungu dan memberikan pengalaman baru dalam mengenali bahasa isyarat melalui perangkat Android. Namun, penelitian ini memiliki beberapa batasan: hanya fokus pada perancangan aplikasi berbasis Android menggunakan Jetpack Compose, *model deep learning* yang digunakan merupakan hasil penelitian sebelumnya tanpa dilakukan pengembangan ulang, serta pendeteksian bahasa isyarat dibatasi pada alfabet statis SIBI, tidak mencakup huruf z dan j yang memerlukan gerakan dinamis.

#### Tunarungu dan Tunawicara

Tunarungu adalah kondisi gangguan pendengaran yang menyebabkan seseorang tidak dapat mendengar dengan jelas atau sepenuhnya, sehingga membutuhkan alat bantu

dingar dan pendidikan khusus[6]. Sementara itu, tunawicara adalah kondisi di mana seseorang tidak mampu berbicara akibat gangguan pada organ pembentuk suara seperti mulut, langit-langit, lidah, atau pita suara, serta dapat disebabkan oleh defisiensi pendengaran dan keterlambatan perkembangan bahasa[7]. Untuk berkomunikasi, penyandang disabilitas pendengaran dan wicara umumnya menggunakan bahasa isyarat yang mengandalkan gerakan tubuh dan ekspresi wajah untuk menyampaikan huruf atau kata.

#### *Rapid Application Development* (RAD)

RAD merupakan model pengembangan perangkat lunak yang bersifat inkremental dengan durasi yang relatif cepat, bertujuan mempercepat iterasi dalam pengembangan sehingga mempercepat tim mendapatkan evaluasi [8]. RAD merupakan modifikasi dari model pengembangan yang sudah ada sebelumnya yaitu *waterfall*, namun memiliki fleksibilitas terhadap perubahan. Pendekatan ini menekankan pada pembuatan *prototype* secara interaktif, sehingga pengguna dapat terlibat langsung pada proses pengujian [9].

#### *User Acceptance Test* (UAT)

*User Acceptance Test* (UAT) adalah metode pengujian yang melibatkan pengguna akhir untuk memastikan bahwa perangkat lunak yang dikembangkan sudah sesuai dengan kebutuhan dan harapan mereka. Dalam pengujian ini, pengguna menjalankan skenario yang telah ditetapkan dan memberikan umpan balik terkait kinerja dan kesesuaian sistem. Hasil dari UAT menjadi dasar penilaian untuk menentukan apakah perangkat lunak layak diluncurkan secara resmi [10].

#### Skala Likert

Skala Likert adalah metode pengukuran yang digunakan untuk mengevaluasi opini, sikap, atau pernyataan responden terhadap topik tertentu, terutama dalam penelitian sosial dan perilaku. Skala ini dikembangkan oleh Reinsis Likert dan bertujuan memberikan analisis kuantitatif atas tingkat persetujuan responden terhadap suatu pernyataan. Umumnya, responden diminta menilai tingkat persetujuan berdasarkan pandangan pribadi mereka. Hasil dari penilaian tersebut kemudian dihitung dan dianalisis untuk memperoleh bobot atau skor yang merepresentasikan pandangan responden. Keunggulan skala ini terletak pada fleksibilitasnya, memungkinkan pengukuran tidak hanya pada aspek positif dan negatif, tetapi juga pada tingkat ketidakpastian atau sikap netral

#### Kotlin

Kotlin adalah bahasa pemrograman yang dikembangkan oleh perusahaan asal Rusia bernama JetBrains serta nama Kotlin sendiri berasal dari nama kota di Rusia. Bahasa ini dirancang untuk mempermudah developer Java dengan menghadirkan bahasa pemrograman yang ringkas dan

modern namun tetap memiliki basis yang sama yaitu Java membuat bahasa ini sudah memiliki fondasi yang matang [11].

### Jetpack Compose

Jetpack Compose merupakan pengembangan dari *framework* UI pada proyek Android *Native* yang direkomendasikan oleh tim Android Google, *framework* ini mempercepat pembuatan UI dengan pendekatan deklaratif sehingga mengurangi kode yang tidak diperlukan [12]. Jetpack Compose menerapkan paradigma *functional programming*, dimana setiap fungsi ditandai dengan anotasi `@Compose` [13]. Sebelum hadirnya Jetpack Compose, pengembangan UI di proyek android menggunakan *view system* yaitu memisahkan kode UI dengan logika menyebabkan pengembangan lebih lama dan berpotensi menimbulkan *bug*.

### CameraX

CameraX Merupakan salah satu dari kumpulan *library* Jetpack yang dirancang untuk mengembangkan fitur kamera pada aplikasi Android menjadi lebih interaktif, dengan mengedepankan kesederhanaan dan mampu beradaptasi dengan berbagai API Android [14]. Sebelum adanya CameraX developer diharuskan menggunakan API bawaan perangkat android yang memiliki kualitas gambar kurang jelas.

### Tensorflow Lite

TensorFlow Lite merupakan *framework open-source* yang dirancang khusus untuk kebutuhan *model machine learning* pada perangkat dengan sumber daya terbatas, seperti Android dan iOS. *Framework* ini menawarkan solusi yang ringan dan cepat, sehingga memungkinkan pengembangan dan penerapan *model machine learning* secara langsung di perangkat *mobile*. Dengan menggunakan TensorFlow Lite, proses integrasi dan implementasi *deep learning* menjadi lebih sederhana dan efisien [15].

### Model Deep Learning

Implementasi *model deep learning* pada penelitian ini menggunakan pendekatan *Deep Learning* dan *Transfer Learning* untuk membangun sistem klasifikasi huruf statis SIBI. *Deep Learning* adalah cabang dari kecerdasan buatan yang memungkinkan komputer mempelajari pola dari data dan membuat keputusan tanpa harus diprogram secara eksplisit [2]. Teknologi ini telah banyak dimanfaatkan di berbagai bidang seperti kesehatan, industri, dan teknologi [15]. Pengembangan *model deep learning* memerlukan proses pelatihan (*training*) dengan dataset yang mewakili objek yang ingin dikenali [5]. Dalam penelitian ini, *deep learning* digunakan untuk memproses citra tangan dan mengekstraksi fitur secara otomatis melalui jaringan saraf dengan banyak lapisan (*multi-layer*) .

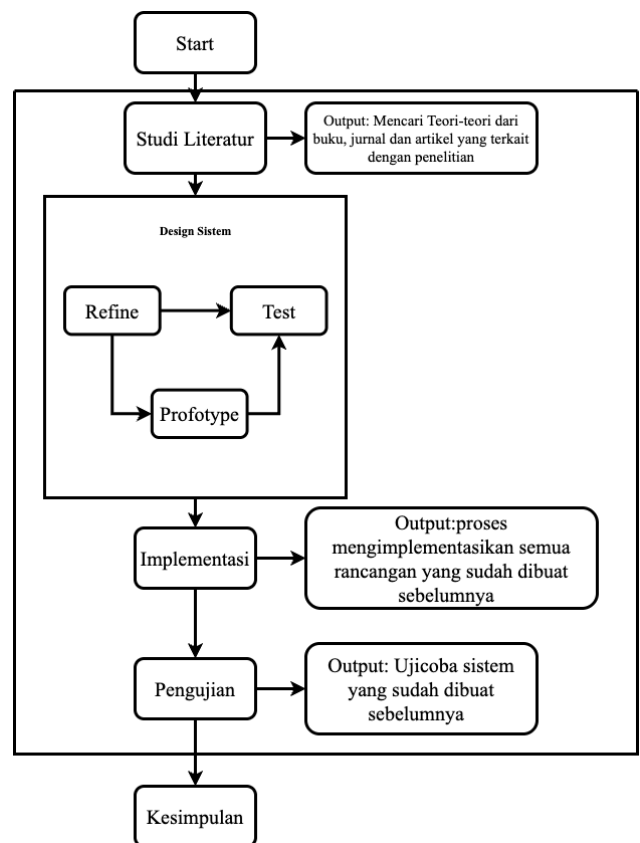
Untuk meningkatkan efisiensi dan akurasi, penelitian ini menerapkan *transfer learning* dengan memanfaatkan arsitektur MobileNetV2 yang sudah dilatih sebelumnya [5]. MobileNetV2 dipilih karena merupakan model *convolutional neural network* (CNN) yang ringan dan dirancang khusus untuk perangkat dengan sumber daya terbatas seperti *smartphone*, sehingga cocok untuk aplikasi Android. Model yang sudah dilatih kemudian dikonversi ke format TensorFlow Lite (.tflite) agar dapat diintegrasikan langsung ke dalam aplikasi dan berjalan secara *on-device* tanpa koneksi internet. Dengan pendekatan ini, proses deteksi bahasa isyarat dapat dilakukan secara *realtime* dengan latensi rendah dan penggunaan sumber daya yang efisien.

## 2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan menyebarkan kuesioner kepada responden dalam pengumpulan data dan analisis data.

### 2.1 Tahapan Penelitian

Dalam proses penelitian terdapat serangkaian tahapan, yang harus dilakukan, tahapan tersebut dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

#### a. Studi Literatur

Pada tahap studi literatur, peneliti meninjau berbagai jurnal terkait pengembangan aplikasi pendeteksi bahasa isyarat, khususnya yang menggunakan *Deep Learning* pada

platform *mobile* dan *web*. Peneliti mempelajari arsitektur model, teknik *transfer learning*, dan metode evaluasi yang umum digunakan. Selain itu, peneliti juga mendalami dokumentasi resmi Google Android tentang Kotlin, CameraX dan Jetpack Compose untuk menentukan struktur aplikasi dan alur data yang sesuai. Hasil kajian ini menjadi dasar dalam perancangan kerangka konseptual, pemilihan teknologi, dan metodologi pengembangan aplikasi.

#### b. Design Sistem

Pada tahapan ini, peneliti melakukan analisis dan perancangan arsitektur sistem berdasarkan informasi yang telah dikumpulkan sebelumnya. Proses ini mencakup perancangan UI/UX, pemilihan *design pattern*, penentuan *library* dan *plugin* yang akan digunakan, serta penetapan fitur-fitur yang akan dikembangkan. Hasil dari tahapan ini menjadi dasar penting dalam proses pembangunan aplikasi.

#### c. Implementasi

Setelah tahapan desain sistem, peneliti mulai mengimplementasikan hasil yang telah dirancang ke dalam bahasa pemrograman Kotlin. Fokus pada tahapan ini mencakup *slicing* desain UI/UX, implementasi *model Deep Learning*, pengembangan algoritma, serta pembuatan fitur-fitur yang dibutuhkan.

#### d. Pengujian

Pengujian dilakukan untuk memastikan aplikasi yang telah dirancang dapat berjalan sesuai kebutuhan tanpa terjadi *crash* atau *error*. Peneliti melakukan pengujian dengan metode *unit test* untuk menguji setiap unit kecil dalam alur kerja aplikasi, serta menggunakan *Instrumentation UI Test* untuk memastikan performa antarmuka berjalan dengan baik. Selain itu, peneliti juga melakukan pengujian manual pada implementasi model *deep learning* yang dijalankan secara *realtime* melalui kamera pada aplikasi.

#### e. Kesimpulan

Pada tahap ini, peneliti menyimpulkan hasil pengembangan dan pengujian yang telah dilakukan. Peneliti juga memaparkan solusi yang diimplementasikan untuk membantu orang normal berkomunikasi dengan penyandang tunarungu atau tunawicara.

### 2.2 Metode Pengumpulan Data, Instrumen Penelitian, dan Metode Pengujian

Penelitian ini menggunakan pendekatan *Research and Development (R&D)* yang berfokus pada pengembangan aplikasi penerjemah bahasa isyarat SIBI berbasis Android. Pengumpulan data dilakukan melalui angket (kuesioner) yang disebar secara daring kepada responden umum. Kuesioner ini bertujuan untuk mengidentifikasi kebutuhan pengguna, memahami permasalahan dalam komunikasi dengan penyandang tunarungu, serta mengevaluasi penerimaan terhadap aplikasi yang dikembangkan.

Teknik pengambilan sampel menggunakan *convenience sampling*, yaitu memilih responden yang mudah dijangkau dan bersedia mengikuti pengujian. Jumlah responden minimal 30 orang sesuai standar penelitian eksploratif untuk memastikan hasil yang cukup representatif.

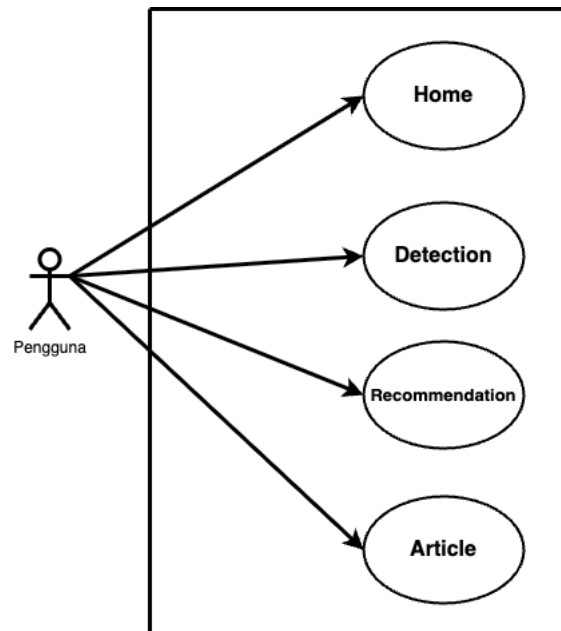
Metode pengujian dalam penelitian ini melibatkan dua tahap, yaitu *Black Box Testing* dan *User Acceptance Test (UAT)*. *Black Box Testing* digunakan untuk menguji fungsionalitas aplikasi berdasarkan skenario yang telah ditentukan tanpa memeriksa struktur internal. Sedangkan UAT melibatkan pengguna akhir untuk mengevaluasi apakah aplikasi sudah sesuai dengan kebutuhan dan layak digunakan. Hasil pengujian dari UAT menjadi acuan dalam menilai tingkat penerimaan dan kelayakan aplikasi.

### 3. HASIL DAN PEMBAHASAN

Pada tahap ini, dilakukan identifikasi kebutuhan pengguna melalui kuesioner untuk memahami kendala dalam komunikasi dengan penyandang tunarungu dan harapan terhadap aplikasi. Hasilnya menunjukkan bahwa pengguna membutuhkan aplikasi penerjemah bahasa isyarat SIBI yang akurat, mudah digunakan, dan memiliki tampilan sederhana untuk mendukung interaksi sehari-hari.

#### 3.1 Usecase Diagram

Pada bagian ini akan dijelaskan tentang *Usecase diagram* yang bertujuan untuk menggambarkan pengguna dengan aplikasi, pada Gambar 2 berikut ini.

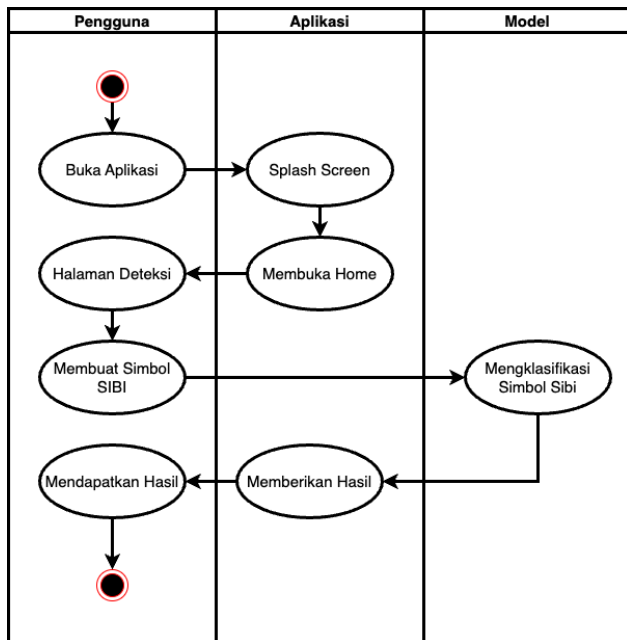


Gambar 2. Usecase Diagram

*Usecase diagram* di atas menggambarkan interaksi antara aktor dan sistem dalam aplikasi pendeteksi Bahasa isyarat SIBI. Aktor “Pengguna” dapat mengakses empat halaman utama yaitu *home*, *detection*, *recommendation* dan *article*.

### 3.2 Activity Diagram

Activity Diagram pada aplikasi pendeteksi bahasa isyarat SIBI menggambarkan alur aktivitas utama yang melibatkan pengguna. Proses dalam diagram melibatkan setiap aktor, sistem aplikasi dan model.



Gambar 3. Activity Diagram

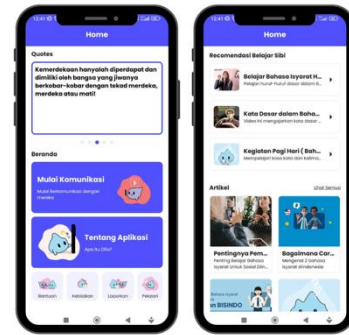
Pada Gambar 3, diperlihatkan alur kerja aplikasi saat pengguna melakukan deteksi Bahasa isyarat SIBI, mulai dari membuka aplikasi lalu membuka halaman deteksi, selanjutnya pengguna membuat simbol huruf SIBI, dan model akan membaca dan analisis simbol serta memberikan hasilnya ke pengguna secara *realtime*.

### 3.3 User Interface

Dalam tahap ini, adalah proses perancangan aplikasi pendeteksi Bahasa isyarat SIBI berbasis android, mencakup beberapa halaman. Gambar 4 menunjukkan halaman *onboarding*, Gambar 5 menunjukkan halaman *home screen*, dan Gambar 6 menunjukkan halaman deteksi.



Gambar 4. Onboarding



Gambar 5. Home Screen



Gambar 6. Deteksi SIBI Screen

### 3.4 Hasil Pengujian

Ditahap ini akan dilakukan 2 pengujian menggunakan *black box testing* dan *user acceptance testing* berbasis skala likert.

#### 3.4.1 Black Box Testing

Tabel 1. Black Box Testing

No	Pengujian	Ekspektasi	Hasil
1	Splash Screen Pertama Kali	Menampilkan <i>Splash Screen</i> dan navigasi ke halaman <i>Onboarding</i>	Berhasil
2	Splash Screen	Menampilkan <i>Splash Screen</i> dan navigasi ke halaman <i>Home</i>	Berhasil
3	Onboarding (Slide)	Menampilkan <i>Screen</i> tentang informasi aplikasi	Berhasil
4	Onboarding	Navigasi Ke halaman <i>Get Started</i>	Berhasil
5	Get Started	Navigasi Kehalaman <i>home</i> dan ubah status	Berhasil
6	Home Screen Quotes	Aplikasi menampilkan <i>quotes</i> lain berdasarkan arah <i>slide</i> pengguna	Berhasil
7	Home Screen Deteksi	Aplikasi Menampilkan Halaman Deteksi	Berhasil
8	Home Screen Rekomemndation	Aplikasi berhasil Menampilkan <i>recomemndation</i>	Berhasil
9	Home Screen Tentang aplikasi	Aplikasi berhasil pindah ke halaman tentang aplikasi	Berhasil

No	Pengujian	Ekspektasi	Hasil
10	Home Screen - Tentang aplikasi	Aplikasi berhasil pindah ke halaman tentang aplikasi	Berhasil
11	Home Screen - Article	Aplikasi menampilkan yang dipilih	Berhasil
12	Deteksi SIBI-Permission	Aplikasi berhasil membuka kamera	Berhasil
13	Deteksi SIBI -Flash	Perangkat berhasil menyalakan flash	Berhasil
14	Deteksi SIBI - Ubah Kamera	Aplikasi berhasil mengubah kamera menjadi kamara depan	Berhasil
15	Deteksi SIBI - Deteksi Bahasa Isyarat SIBI	Menampilkan Huruf SIBI beserta skor klasifikasinya	Berhasil
16	Tentang Aplikasi	Aplikasi dapat menampilkan deskripsi tentang aplikasi berdasarkan yang dipilih	Berhasil

Terdapat rumus untuk menghitung keberhasilan dan kegagalan pada pengujian *black box testing*.

Persentase = (Jumlah skenario berhasil / jumlah skenario) x 100%

Berdasarkan data pada Tabel 1 didapati hasil persentase sebagai berikut:

Persentase = (16/16) x 100% = 100%

Dengan data tersebut menandakan bahwa aplikasi dapat berjalan sesuai dengan harapan pengujian.

### 3.4.2 User Acceptance Test

Metode pengambilan data pada pengujian ini menggunakan skala likert dengan menyebarkan kuesioner kepada responden menggunakan Google Form, dengan detail sebagai berikut:

a) Total Responden sebanyak 25 responden, analisis pengujian ini dihitung menggunakan skala likert yang memiliki kriteria berikut.

1) 1 = Sangat Tidak Setuju (STS)

2) 2 = Tidak Setuju (TS)

3) 3 = Kurang Setuju (KS)

4) 4 = Setuju (S)

5) 5 = Sangat Setuju (SS)

b) Rumus Perhitungan Nilai rata-rata dan persentase sebagai berikut:

1) Nilai rata-rata = (jumlah bobot / jumlah pertanyaan x jumlah responden)

2) Persentase = (nilai rata-rata / skor maksimum) x 100%

c) Terdapat 14 Pertanyaan, dibagi menjadi 4 kelompok, setiap bagian memiliki 3 sampai 5 pertanyaan, dengan detail sebagai berikut:

Tabel 2. User Acceptance Test

No	Pertanyaan	STS*1	TS*2	KS*3	S*4	SS*5	Bobot
1	Apakah Aplikasi ini mudah digunakan?	-	-	3	24	90	117
2	Navigasi Antar halaman dalam aplikasi terasa intuitif?	-	2	6	20	85	113
3	Saya tidak mengalami kesulitan dalam memahami fungsi dari setiap tombol / menu?	-	-	9	24	80	113
<b>Kemudahan Pengguna</b>							<b>343</b>
4	Tampilan Aplikasi Menarik Secara Penampilan?	-	-	3	12	105	120
5	Pemilihan Ikon, Warna dan tata letak memudahkan pengguna aplikasi?	-	2	-	32	80	114
6	Informasi pada aplikasi mudah dipahami?	-	-	9	16	90	115
<b>Desain Antar Muka</b>							<b>349</b>
7	Fitur Penerjemah bahasa isyarat bekerja dengan baik?	1	2	-	32	80	115
8	Aplikasi Dapat Mendeteksi Huruf-huruf bahasa isyarat sibi statis (a s.d z kecuali j dan z) dengan akurat?	2	2	9	32	55	100
9	Proses Pendeteksian gerakan ke <i>text</i> berlangsung secara cepat dan <i>realtime</i> ?	1	-	-	28	85	114
10	Aplikasi Berjalan dengan lancar tanpa <i>force close/ error</i> ?	1	-	3	16	95	115
11	Waktu Respons aplikasi terhadap <i>input</i> pengguna tergolong cepat?	1	-	3	28	80	112
<b>Fungsionalitas &amp; Performa</b>							<b>556</b>
12	Saya merasa puas menggunakan aplikasi ini?	-	2	-	28	85	115

No	Pertanyaan	STS* 1	TS* 2	KS* 3	S* 4	SS* 5	Bobot
13	Aplikasi ini dapat membantu untuk mendeteksi bahasa isyarat sibi?	-	-	3	28	85	116
14	Saya bersedia menggunakan kembali aplikasi ini dimasa mendatang?	-	-	3	16	100	119
<b>Kepuasan Pengguna</b>							<b>350</b>
<b>Total</b>							<b>1598</b>

Pada Tabel 2, hasil bobot di setiap pertanyaan dilakukan pengelompokan, kemudian dilakukan perhitungan nilai rata-rata dan persentase keseluruhan.

Tabel 3. Perhitungan UAT

No	Kelompok	Nilai Rata-rata	Persentase
1	Kemudahan Pengguna	4,6	91%
2	Desain Antar Muka	4,7	93%
3	Fungsionalitas & Performa	4,4	86%
4	Total Keseluruhan Evaluasi	4,6	91%

Hasil dari keseluruhan pertanyaan UAT dapat dilihat pada Tabel 3 di atas, semua data dilakukan penjumlahan lalu dilakukan perhitungan persentase dengan hasil sebesar 91%.

### 3.5 Hasil Evaluasi

Evaluasi hasil didasarkan pada perhitungan total skor tanggapan responden terhadap pernyataan. Total skor diperoleh dari akumulasi tanggapan "Sangat Setuju" ( $239 \times 5 = 1195$ ), "Setuju" ( $84 \times 4 = 336$ ), "Kurang Setuju" ( $17 \times 3 = 51$ ), "Tidak Setuju" ( $5 \times 2 = 10$ ), dan "Sangat Tidak Setuju" ( $6 \times 1 = 6$ ), menghasilkan total skor 1598.

Untuk memastikan akurasi, perhitungan nilai tertinggi dan terendah dilakukan. Dengan jumlah tanggapan 350, skor tertinggi (skor maksimum 5) adalah  $350 \times 5 = 1750$ , dan skor terendah (skor minimum 1) adalah  $350 \times 1 = 350$ .

Interpretasi penilaian responden terhadap ekspektasi aplikasi pendeteksi bahasa isyarat SIBI dihitung menggunakan Persentase Indeks sebagai berikut:

$$\text{Persentase Indeks} = (\text{Total Skor} / \text{Nilai Tertinggi}) \times 100\%$$

Dengan nilai-nilai yang diperoleh:

- Persentase Indeks =  $(1598/1750) \times 100\%$
- Persentase Indeks = 91,3%

Nilai 91,3% pada pengujian *User Acceptance Test* (UAT) dianggap sebagai penilaian positif. Oleh karena itu, disimpulkan bahwa fungsionalitas aplikasi dapat memenuhi kebutuhan dan harapan pengguna, meskipun beberapa

fungsionalitas dan kinerja memerlukan peningkatan agar berjalan lancar di berbagai perangkat Android.

## 4. KESIMPULAN

Berdasarkan hasil evaluasi dan implementasi dapat disimpulkan bahwa Aplikasi pendeteksi bahasa isyarat SIBI berbasis Android berhasil dikembangkan menggunakan *Jetpack Compose dan Clean Architecture*, dengan integrasi *model deep learning* melalui TensorFlow Lite. Proses pengembangan dilakukan secara sistematis, mulai dari analisis kebutuhan hingga implementasi dan integrasi model.

Pengujian *Black Box* menunjukkan keberhasilan 100% pada 16 skenario, sedangkan *User Acceptance Test* (UAT) oleh 25 responden menghasilkan tingkat kepuasan sebesar 91,3%. Hasil ini menunjukkan bahwa aplikasi memiliki antarmuka yang intuitif dan kinerja yang stabil.

Penelitian ini melanjutkan studi sebelumnya yang hanya berfokus pada pengembangan model tanpa implementasi aplikasi. Dengan mengonversi model ke format .flite, aplikasi berhasil dijalankan secara efisien di perangkat android.

Secara keseluruhan, aplikasi ini berfungsi dengan baik dalam mengenali huruf statis SIBI dan berpotensi mendukung komunikasi inklusif bagi penyandang tunarungu. Masukan pengguna menjadi dasar penting untuk pengembangan lebih lanjut.

## DAFTAR PUSTAKA

- [1] R. Fatmawati, R. Asmara, Y. R. Prayogi, and R. Y. Hakkun, "Aplikasi Pembelajaran Sistem Isyarat Bahasa Indonesia (SIBI) Berbasis Voice Menggunakan OpenSIBI," *Technomedia Journal*, vol. 7, no. 1, pp. 22–39, 2022, doi: 10.33050/tmj.v7i1.1690.
- [2] S. Apendi, C. Setianingsih, and M. W. Paryasto, "Deteksi Bahasa Isyarat Sistem Isyarat Bahasa Indonesia Menggunakan Metode Single Shot Multibox Detector," *eProceedings of Engineering*, vol. 10, no. 1, pp. 249–255, 2023, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/19322>
- [3] World Health Organization (WHO), *World Report On Disability*. Jenewa, Swiss: WHO, 2021.
- [4] S. F. T. Andalas, "209,3 Juta Orang di Indonesia Menggunakan Smartphone pada Tahun 2023," Good stats. Accessed: Sep. 24, 2024. [Online]. Available: <https://data.goodstats.id/statistic/2093-juta-orang-di-indonesia-menggunakan-smartphone-pada-tahun-2023-cbha0>

- [5] F. P. Al Fajri, "Perancangan Model Deep Learning untuk Penerjemah Bahasa Isyarat SIBI menggunakan Transfer Learning MobileNetV2," Tugas Akhir Sekolah Tinggi Teknologi Terpadu Nurul Fikri, Depok, 2024.
- [6] D. I. Setyawan, H. Tolle, and A. P. Kharisma, "Perancangan Aplikasi Communication Board Berbasis Android Tablet Sebagai Media Pembelajaran dan Komunikasi Bagi Anak Tuna Rungu," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 8, pp. 2933–2943, 2018, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [7] N. Wiranda and A. E. Putra, "Model Identifikasi Kata Ucapan Tuna Wicara," *Indonesian Journal of Electronics and Instrumentation Systems (IJEIS)*, vol. 9, no. 2, pp. 131–140, 2019.
- [8] W. D. Prastowo, D. Danianti, and A. Pramuntadi, "Analisis Risiko Pada Pengembangan Perangkat Lunak Menggunakan Metode Agile Dan Rad (Rapid Application Development)," *Citizen : Jurnal Ilmiah Multidisiplin Indonesia*, vol. 3, no. 3, pp. 169–174, 2023, doi: 10.53866/jimi.v3i3.388.
- [9] D. Murdiani and M. Sobirin, "Perbandingan Metodologi Waterfall Dan RAD Dalam Pengembangan Sistem Informasi," *JINTEKS (Jurnal Informatika Teknologi dan Sains)*, vol. 4, no. 4, pp. 302–306, 2022, [Online]. Available: <http://www.jurnal.uts.ac.id/index.php/JINTEKS/article/view/2008>
- [10] E. Susanti, "Implementasi RESTful API dalam Pembuatan Master Data Planogram Menggunakan Framework Flask (Studi Kasus: PT Sumber Alfaria Trijaya, Tbk)," *Techno.Com*, vol. 19, no. 3, pp. 295–307, 2020, doi: 10.33633/tc.v19i3.3468.
- [11] JetBrains, "Get started with Kotlin," *Kotlin Language*. Accessed: Sep. 29, 2024. [Online]. Available: <https://kotlinlang.org/docs/getting-started.html#backend>
- [12] F. Dzulqarnain and T. Tukino, "Rancang Bangun Aplikasi Belajar Arab Untuk Android Menggunakan Jetpack Compose Dan Kotlin," *Computer Based Information System Journal*, vol. 11, no. 1, pp. 25–35, 2023, doi: 10.33884/cbis.v11i1.6666.
- [13] E. Milla and M. Radonjić, "Analysis of Developing Native Android Applications Using Xml and Jetpack Compose," *Balkan Journal of Applied Mathematics and Informatics*, vol. 6, no. 2, pp. 167–178, 2023, [Online]. Available: <https://js.ugd.edu.mk/index.php/bjami/article/view/6019>
- [14] M. Peleš, S. Jevremović, A. Simović, and A. Hadžić, "Possibilities for developing and implementing a mobile application for recognizing the shape of the environment, text, and reading QR codes using the Android CameraX framework and the Machine Learning Kit," *Procedia of Economics and Business Administration*, vol. 12, no. 2, pp. 119–127, 2021, [Online]. Available: [www.icesba.euhttps://doi.org/10.26458/v6.i1.x](http://www.icesba.euhttps://doi.org/10.26458/v6.i1.x)
- [15] S. S. Sindarto, D. E. Ratnawati, and I. Arwani, "Klasifikasi Citra Sistem Isyarat Bahasa Indonesia (SIBI) dengan Metode Convolutional Neural Network pada Perangkat Lunak berbasis Android," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 5, pp. 2129–2138, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>