



RANCANG BANGUN APLIKASI PENERJEMAH BAHASA ISYARAT BERBASIS MEDIAPIPE SOLUTION

Kelompok Limanyala :

- Vidi Septri Argalus MP 2107411032
- Muhammad Fahmi Zuhdi 2107411033
- Muhammad Riziq Ramadhan 2107411035
- Rhaditya Hadi Wibowo 2107411047

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA**

2024

Abstract

Perkembangan teknologi informasi dan komunikasi membuka peluang baru dalam meningkatkan aksesibilitas dan inklusivitas bagi penyandang disabilitas. Penelitian ini bertujuan untuk mengembangkan aplikasi penerjemah bahasa isyarat yang memanfaatkan teknologi AI dan MediaPipe Solution untuk memfasilitasi komunikasi antara individu yang tidak menguasai bahasa isyarat dan penyandang gangguan pendengaran. Aplikasi ini mengintegrasikan teknologi pengenalan gerakan tangan dengan machine learning untuk menerjemahkan bahasa isyarat ke dalam bentuk teks secara real-time, sehingga memudahkan interaksi dan komunikasi yang lebih efektif.

Pemilihan bentuk badan usaha untuk proyek ini adalah Perseroan Terbatas (PT) untuk memastikan tanggung jawab terbatas, keberlanjutan usaha, kemudahan dalam menggalang modal, dan citra profesional. Proses legalitas melibatkan langkah-langkah seperti pengajuan nama PT, pembuatan akta pendirian, SKDP, NPWP, dan SIUP, serta pendaftaran Tanda Daftar Perusahaan.

Kajian pustaka menunjukkan bahwa teknologi pengenalan gerakan tangan berbasis machine learning dan framework MediaPipe Solution memberikan solusi efektif untuk deteksi gerakan dan pose tubuh. Penelitian sebelumnya menggarisbawahi potensi besar teknologi ini dalam meningkatkan akurasi dan efisiensi penerjemahan bahasa isyarat.

Proyek ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan teknologi penerjemahan bahasa isyarat, memperbaiki aksesibilitas komunikasi, dan menjadi model untuk inovasi teknologi dalam sektor komunikasi. Selain itu, proyek ini juga bertujuan untuk mendukung penyandang disabilitas dengan menyediakan alat komunikasi yang lebih mudah diakses dan efisien.

DAFTAR ISI

Abstract.....	1
DAFTAR ISI.....	2
Daftar Tabel.....	5
Daftar Gambar.....	6
BAB I.....	8
1.1 Latar Belakang Masalah.....	8
1.2 Rumusan Masalah.....	9
1.3 Batasan Masalah.....	10
1.4 Tujuan Penelitian.....	10
1.5 Manfaat Penelitian.....	11
BAB II.....	12
2.1 Bahasa Isyarat.....	12
2.2 Tunarungu.....	12
2.3 Machine Learning.....	13
2.4 Computer Vision.....	14
2.5 Mediapipe.....	15
2.6 LSTM (Long Short-Term Memory).....	17
2.7 Python.....	17
2.8 Google Firestore.....	18
2.9 Aplikasi Mobile.....	18
2.10 Penelitian Sejenis.....	18
BAB III.....	20
3.1 Rancangan Penelitian.....	20
3.2 Tahapan Penelitian.....	21
3.3 Objek Penelitian.....	23
BAB IV.....	24
4.1 Analisis Kebutuhan.....	24
4.1.1 Kebutuhan Fungsional.....	24
4.1.2 Kebutuhan Non-Fungsional.....	25
4.1.3 Kebutuhan Pengguna.....	26
4.2 Perancangan Aplikasi.....	27
4.2.1 Flowchart.....	28
4.2.2 Use Case Diagram.....	29
4.2.3 Activity Diagram.....	30
4.3 Implementasi Sistem.....	34
4.3.1 Halaman Login.....	35

4.3.2 Halaman Utama.....	36
4.3.3 Letter Level Gesture Recognition.....	36
4.3.3.1 Load Dataset.....	37
4.3.3.2 Model.....	40
4.3.4 Word Level Gesture Recognition.....	42
4.3.4.1. Data Extraction dan Feature Extraction.....	42
4.3.4.2. Data Preparation.....	43
4.3.4.3. Model Preparation.....	43
4.3.5 Speech-to-Text Recognition.....	45
4.3.6 Hate Speech Detection.....	46
4.3.6.1 Data Preproccesing.....	46
4.3.6.2 Model Preperation.....	48
4.3.6 Halaman Profile.....	51
4.3.7 Halaman Kamus.....	52
4.4 Pengujian.....	53
4.4.1 Deskripsi Pengujian.....	53
4.4.2 Prosedur Pengujian.....	54
4.4.2.1 Blackbox Testing.....	54
4.4.2.2 Analisis Model Machine Learning.....	55
4.4.3 Data Hasil Pengujian.....	56
4.4.3.1 Blackbox Testing.....	56
4.4.3.2 Analisis Model Machine Learning.....	59
BAB V.....	65
5.1 Simpulan.....	65
5.2 Saran.....	65
DAFTAR PUSTAKA.....	67

Daftar Tabel

Tabel 4.1 Prosedur Pengujian Blackbox Testing.....	40
Tabel 4.2 Prosedur Pengujian WhiteBox Testing.....	41

Daftar Gambar

Gambar 2.1 Fitur pada Mediapipe.....	9
Gambar 4.1 Flowchart Diagram.....	19
Gambar 4.2 Use Case Diagram.....	20
Gambar 4.3 Activity Diagram Gesture Recognition (Word Level).....	21
Gambar 4.4 Activity Diagram Gesture Recognition (Letter Level).....	22
Gambar 4.5 Activity Diagram Speech-to-Text.....	23
Gambar 4.7 Tampilan UI Halaman Utama.....	25
Gambar 4.8 Examples dataset.....	26
Gambar 4.9 Split Data.....	27
Gambar 4.10 Model summary().....	27
Gambar 4.11 Hasil Implementasi Model di Aplikasi Mobile.....	29
Gambar 4.12 Hasil.....	31
Gambar 4.13 Grafik Analisis per-Epoch.....	32
Gambar 4.14 Hasil Implementasi Model di Aplikasi Mobile.....	32
Gambar 4.15 Tampilan UI Fitur Speech-to-Text.....	33
Gambar 4.16 Hasil Text Cleaning.....	34
Gambar 4.17 Hasil Text Stemming.....	34
Gambar 4.18 Hasil Converting Slang Word.....	35
Gambar 4.19 Hasil Split Dataset.....	35
Gambar 4.20 Hasil Tokenizing.....	36
Gambar 4.21 Hasil Training Model.....	36
Gambar 4.22 Tampilan UI Halaman Profile.....	37
Gambar 4.23 Tampilan UI Halaman Kamus.....	38

Gambar 4.24 Tampilan UI Halaman Kamus Berdasarkan Hasil Pencarian.....	39
Gambar 4.25 Hasil Akurasi per Epoch.....	42
Gambar 4.26 Grafik Analisis per-Epoch.....	42
Gambar 4.27 Hasil Confussion Matrix.....	43
Gambar 4.28 Hasil Matrix Evaluasi.....	44
Gambar 4.29 Matrix Evaluasi Model Hate Speech Detection.....	44

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Bahasa isyarat adalah metode komunikasi penting bagi individu dengan gangguan pendengaran atau tunarungu, yang memungkinkan mereka berinteraksi sosial dengan masyarakat. Berdasarkan data dari Kementerian Pendidikan dan Kebudayaan (*Belajar Bahasa Isyarat*, 2023) yang merujuk pada Kementerian Sosial mencatat bahwa jumlah penyandang disabilitas di Indonesia sekitar 30,38 juta jiwa. Dari jumlah tersebut 7,03 persen merupakan disabilitas tuli dan tunarungu. Sayangnya masih banyak teman tunarungu yang masih kesulitan dalam berkomunikasi dengan masyarakat umum, hal ini menimbulkan adanya *language barrier* antara kedua kelompok masyarakat, yang dapat berdampak pada minimnya aksesibilitas, kualitas layanan, dan kesempatan kerja bagi penyandang disabilitas serta menghambat mereka untuk berpartisipasi penuh dalam masyarakat.

Contoh nyata kesenjangan komunikasi antara lain tunarungu seringkali menghadapi kesulitan dalam mengakses layanan publik seperti rumah sakit, bank, dan kantor pemerintahan karena kurangnya tenaga profesional yang menguasai bahasa isyarat. Kurangnya guru dan tenaga pendidik yang menguasai bahasa isyarat di sekolah umum membuat anak tunarungu kesulitan mengikuti pelajaran dan berinteraksi dengan teman sebayanya. Kesempatan Kerja: Kesulitan berkomunikasi menjadi penghalang bagi tunarungu untuk mendapatkan pekerjaan yang layak.

Dengan kemajuan teknologi informasi dan komunikasi membuka peluang baru untuk mengatasi kesenjangan bahasa antara masyarakat tunarungu dan non-disabilitas. Teknologi yang dapat dimanfaatkan adalah *machine learning*, khususnya konsep *computer vision* dalam bidang pengenalan gestur. Salah satu upaya yang dapat dilakukan dengan mengembangkan aplikasi penerjemah bahasa isyarat yang memainkan peran penting dalam menjembatani kesenjangan komunikasi antara komunitas tuna rungu dan masyarakat umum.

Proposal penelitian ini menggunakan desain dan penerapan aplikasi terjemahan bahasa isyarat yang memanfaatkan kerangka MediaPipe Solution untuk mengenali dan menafsirkan isyarat bahasa isyarat secara real-time secara akurat.

Penelitian ini berfokus pada pengembangan dan perancangan sistem penerjemah bahasa isyarat yang akurat dalam mengenali isyarat huruf dan kata dengan bahasa Indonesia, sehingga memungkinkan komunikasi antar komunitas tunarungu dan masyarakat dan dapat meningkatkan kualitas layanan dan kesempatan berpartisipasi penuh bagi tunarungu dalam kelompok masyarakat.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah tersebut, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang model *machine learning* yang dapat mengenali dan memahami bahasa isyarat dengan akurasi tinggi?
2. Bagaimana implementasi model pengenalan gerak tangan ke dalam suatu aplikasi *mobile* agar dapat digunakan?

1.3 Batasan Masalah

Dalam penelitian yang akan dilakukan ini memiliki batasan atau lingkup masalah yang ditetapkan, antara lain:

1. Penggunaan dataset bahasa isyarat berdasarkan SIBI (Sistem Isyarat Bahasa Indonesia) yang sesuai dengan kaidah bahasa Indonesia dan dinormalisasi serta distandarkan sesuai tata bahasa nasional
2. Pengembangan dan pengujian model menggunakan kerangka MediaPipe Solution yang menyediakan pre-trained model dan dapat memodifikasi model berdasarkan data yang di punya
3. Sign Language yang dapat diterima adalah huruf, angka dan kata, dimana kata yang akan dipelajari sekitar 150-250 kata umum yang sering digunakan.
4. Aplikasi penerjemah bekerja secara *real-time* dalam menerjemahkan bahasa isyarat

1.4 Tujuan Penelitian

Tujuan Penelitian ini untuk merancang dan mengembangkan sebuah aplikasi penerjemah bahasa isyarat dengan menggunakan konsep computer vision dalam machine learning, khususnya melalui penerapan MediaPipe Solution. Aplikasi ini diharapkan dapat mengenali dan memahami bahasa isyarat dengan akurasi yang tinggi, sehingga dapat membantu menjembatani komunikasi antara masyarakat tunarungu dan non-disabilitas.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dapat diperoleh dari penelitian ini, antara lain:

1. Memberikan solusi teknologi yang dapat membantu menjembatani komunikasi antara masyarakat tunarungu dan non-disabilitas
2. Meningkatkan aksesibilitas dan pemahaman bahasa isyarat bagi masyarakat non-disabilitas, sehingga dapat meningkatkan kualitas layanan dan kesempatan kerja bagi penyandang disabilitas khususnya tunarungu
3. Memberikan kontribusi dalam bidang penelitian computer vision dan machine learning, khususnya terkait pengenalan dan pemahaman bahasa isyarat

BAB II

TINJAUAN PUSTAKA

2.1 Bahasa Isyarat

Bahasa isyarat bukan sekadar gerakan tangan yang acak. Bahasa isyarat memiliki struktur gramatika, tata bahasa, dan kosakata yang unik. Di Indonesia, bahasa isyarat yang resmi digunakan adalah Sistem Isyarat Bahasa Indonesia (SIBI). SIBI memiliki aturan gramatika yang berbeda dengan bahasa Indonesia lisan, dan menggunakan kombinasi gerakan tangan, ekspresi wajah, dan gerakan tubuh untuk menyampaikan makna. Meskipun disamping itu ada juga BISINDO merupakan bahasa isyarat yang berkembang secara alamiah pada kelompok masyarakat Tuli di Indonesia, sedangkan SIBI merupakan bahasa isyarat yang distandarisasi oleh pemerintah.

2.2 Tunarungu

Tunarungu adalah suatu kondisi atau keadaan seseorang yang indra pendengarannya sangat berkurang atau terganggu sehingga tidak mampu menerima bunyi, suara, atau rangsangan lain melalui pendengaran. Akibat sulitnya perkembangan pendengaran, maka kemampuan bicara dan bahasa penyandang tunarungu juga ikut menurun, sehingga menyebabkan penyandang tunarungu mengalami keterlambatan dan kesulitan dalam hal-hal yang berkaitan dengan komunikasi (Riadi, 2020). Klasifikasi tunarungu mempunyai tiga batasan berdasarkan berapa lama seseorang dapat menggunakan sisa pendengarannya dengan atau tanpa alat bantu dengar, yaitu:

1. **Sulit mendengar:** namun masih dapat menggunakannya sebagai alat atau bentuk pendengaran utama seseorang berbicara. dan mengembangkan keterampilan berbicara.

2. **Tuli:** yaitu mereka yang pendengarannya tidak lagi dapat digunakan sebagai alat utama untuk mengembangkan kemampuan berbicara, namun masih dapat digunakan sebagai pelengkap penglihatan dan sentuhan.
3. **Tuli total:** yaitu mereka yang sudah sama sekali tidak memiliki pendengaran sehingga tidak dapat digunakan untuk menyimak atau mempersepsi dan mengembangkan bicara.

2.3 Machine Learning

Machine learning (ML) adalah cabang dari kecerdasan buatan (AI) yang berfokus pada pengembangan algoritma dan teknik yang memungkinkan komputer untuk belajar dan membuat prediksi atau keputusan berdasarkan data. *Machine learning* juga digunakan untuk mengidentifikasi pola dalam data dan kemudian menggunakan pola tersebut untuk membuat prediksi atau keputusan tanpa diprogram secara eksplisit untuk tugas tersebut (*What Is Machine Learning?*, 2024). Machine learning dibagi menjadi beberapa kategori berdasarkan bagaimana cara mereka menghasilkan suatu keputusan:

1. Supervised Learning

Model pembelajaran ini melakukan prediksi setelah melihat banyak data dengan ada label yang sudah ditentukan sebagai acuan pembelajaran dan kemudian menemukan hubungan antar elemen dalam data dengan label yang menghasilkan jawaban seperti dalam bentuk prediksi (*regression*) atau klasifikasi.

2. Unsupervised Learning

Model pembelajaran ini melakukan pembelajaran tanpa diberikannya label atau acuan yang benar. Tujuan model *Unsupervised Learning* adalah untuk mengidentifikasi pola yang bermakna di antara data. Dengan kata lain, model tidak memiliki petunjuk tentang cara mengkategorikan setiap bagian data, namun harus menyimpulkan aturannya sendiri. Model

Unsupervised Learning yang umum digunakan menggunakan teknik yang disebut clustering.

3. Reinforcement Learning

Model pembelajaran ini membuat prediksi dengan mendapatkan reward atau penalti berdasarkan tindakan yang dilakukan dalam suatu lingkungan. Sistem pembelajaran dalam model ini bertujuan untuk menghasilkan pengetahuan yang menentukan strategi terbaik untuk mendapatkan imbalan maksimal. Biasanya diimplementasikan pada gerakan suatu robot.

4. Generative AI

Generative AI adalah model terbaru pada konsep *Artificial Intelligence* yang membuat suatu konten baru dari *input*-an pengguna. Misalnya, AI ini dapat membuat gambar unik, komposisi musik hingga lelucon. Juga dapat meringkas artikel, menjelaskan cara melakukan tugas, atau mengedit foto. Generative AI dapat mengambil berbagai *input*-an dan menghasilkan berbagai *output*, seperti teks, gambar, audio, dan video.

2.4 Computer Vision

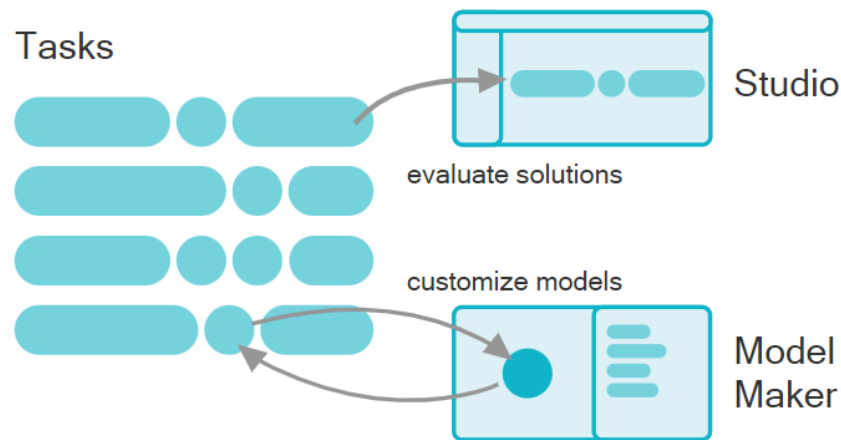
Computer Vision (CV) adalah bidang ilmu komputer yang berfokus pada bagaimana komputer dapat memperoleh, memproses, dan menganalisis data visual dari dunia nyata untuk meniru kemampuan penglihatan manusia. CV menggunakan teknik machine learning untuk mengenali dan memahami gambar dan video. Dengan memadukan machine learning dan computer vision, banyak aplikasi canggih dapat dikembangkan, seperti sistem pengenalan wajah, analisis gambar medis, dan lainnya.

2.5 Mediapipe

Google mengembangkan Mediapipe sebagai *library* dan *tools open source* untuk membantu mempercepat dalam penerapan *machine learning* pada sebuah aplikasi.

Mediapipe dapat digunakan lintas platform memungkinkan pengembang untuk membangun aplikasi yang berjalan dengan lancar di berbagai sistem operasi, termasuk Android, iOS, Linux, dan Windows dalam pengembangannya.

Mediapipe Solution menjadi salah satu bagian dari Mediapipe yang memberikan kemudahan dalam memenuhi kebutuhan aplikasi dengan menyediakan kustomisasi pada kode programnya. Fitur yang diberikan antara lain:



Gambar 2.1 Fitur pada Mediapipe

1. **Mediapipe Task:** dimana adalah *low code API's Cross-platform* dan juga *libraries* yang digunakan untuk men-*deploy*.
2. **Mediapipe Models:** merupakan *pre-trained* model yang dapat digunakan untuk kebutuhan *machine learning*
3. **Mediapipe Model Maker:** digunakan untuk mengkostumisasi model yang disediakan Mediapipe berdasarkan data yang baru
4. **Mediapipe Studio:** studio yang disediakan secara gratis berbasis web oleh Google untuk membantu visualisasi, evaluasi dan menjadi benchmark untuk solution yang dibuat

Pada penelitian ini akan menggunakan task gesture recognition, dimana memungkinkan untuk mengenali gerakan tangan secara real-time, dan memberikan hasil gerakan tangan yang dikenali beserta landmark yang terdeteksi pada tangan. Dengan model ini dapat mengenali gerakan tangan tertentu dari pengguna, dan menjalankan fitur aplikasi yang sesuai dengan gerakan tersebut. Model ini beroperasi pada data gambar dengan model *machine learning*, dan menerima data lalu menghasilkan *landmark* dalam koordinat gambar, penggunaan tangan (tangan kiri/kanan), dan kategori isyarat tangan dari gerakan tangan (*Gesture Recognition Task | Google for Developers, 2024*).

Kemudian juga menggunakan Mediapipe Holistic Landmarker yang pada dasarnya mengkombinasikan komponen dari pose, wajah dan tangan untuk menampilkan landmark yang lebih lengkap. Jika menggunakan seluruh komponen akan menghasilkan outputs dengan total 543 landmark (33 pose landmarks, 468 landmark wajah dan 21 landmark tangan) secara real-time (*Holistic Landmarks Detection Task Guide | Google AI Edge | Google for Developers, 2024*), tetapi untuk memenuhi batasan masalah ini hanya akan mengambil 21 landmark tangan yang digunakan untuk menangkap keypoints dari gerakan.

2.6 LSTM (Long Short-Term Memory)

LSTM merupakan singkatan dari Long Short-Term Memory, yang mana adalah jenis model arsitektur *machine learning* yang dirancang untuk mengatasi permasalahan yang ada pada RNN. Berikut adalah komponen atau arsitektur utama dari LSTM:

1. **Forget Gate:** menerima inputan dan mengatur informasi yang akan dilupakan dari *state*
2. **Input Gate:** menambahkan informasi baru ke *cell state*
3. **Output Gate:** menentukan informasi yang akan dikeluarkan sebagai output

Berdasarkan rujukan dari penelitian bahasa isyarat di India (Goyal & G, 2023, 6), penggunaan LSTM lebih efisien untuk mengenali gerakan yang dinamis dalam mengisyaratkan kata dibandingkan menggunakan CNN dalam pengembangan model *machine learning* berdasarkan akurasi dan tingkat loss yang didapat ketika melakukan pelatihan model. CNN unggul dalam mengenali statis gestur, sedangkan pada penelitian nantinya akan menerapkan gesture recognition yang memiliki tingkat akurasi tinggi.

2.7 Python

Python adalah bahasa pemrograman komputer yang sering digunakan untuk membangun situs web dan perangkat lunak, mengotomatiskan tugas, dan melakukan analisis data. Python dapat digunakan untuk membuat berbagai program berbeda dan tidak dikhususkan untuk masalah tertentu. Fleksibilitas serta kemudahannya bagi pemula, menjadikannya salah satu bahasa pemrograman yang paling banyak digunakan saat ini (*What Is Python Used For?*, 2024). Pertumbuhan minat menggunakan python sangat pesat karena banyaknya kasus yang menggunakan python untuk membangun pemrograman di bidang Teknologi Modern seperti Data Science, AI, Machine Learning dan otomatisasi.

2.8 Google Firestore

Firestore merupakan salah satu layanan yang disediakan oleh Google yang termasuk ke dalam Google Cloud secara khusus pada Google Firebase. Cloud Firestore adalah database berbasis dokumen NoSQL yang memungkinkan untuk menyimpan, menyinkronkan, dan membuat *query* data dengan mudah untuk aplikasi seluler dan web (*Cloud Firestore*, n.d.). Dengan kemudahan yang diberikan yang bersifat serverless dan sinkronisasi data lintas perangkat serta kemudahan dalam penggunaannya dalam membuat database membantu mempercepat pengembangan aplikasi. Firebase juga menyediakan paket gratis dalam layanannya dan produknya termasuk Firestore, sehingga cocok untuk belajar maupun pengembangan skala kecil.

2.9 Aplikasi Mobile

Aplikasi mobile adalah perangkat lunak yang dikembangkan dengan tujuan pengoperasiannya dapat berjalan di perangkat seluler (*smartphone, tablet, iPod, dll*) yang menyediakan informasi dalam format teks dan multimedia, termasuk grafis, audio, dan video. Aplikasi *mobile* disini dirancang untuk memudahkan akses ketika ingin melakukan live stream dalam menerjemahkan bahasa isyarat serta juga menyesuaikan dengan kemudahan integrasi model.

2.10 Penelitian Sejenis

Saat ini sudah ada beberapa literatur yang dapat dijadikan acuan, misalnya penerapan metode CNN pada mendeteksi gerakan tangan statis untuk mengisyaratkan huruf pada bahasa isyarat yang menggunakan pre-trained berjenis *EfficientNetLite4* dengan hasil berupa Tensorflow Lite yang akan dipakai pada *mobile app* (Alfikri, *Penerjemah Bahasa Isyarat*, 2022). Sedangkan untuk pengenalan isyarat berdasarkan kata dapat merujuk pada literatur Indian Sign Language (Goyal & G, 2023, 6) yang mengembangkan pengenalan bahasa isyarat bahasa India melalui Holistic Mediapipe dengan penerapan latihan pembelajaran pada model berdasarkan dataset isyarat India.

Namun pada penelitian ini akan berfokus pada implementasi Mediapipe dimana dapat membantu meningkatkan akurasi model serta juga dapat menerjemahkan isyarat kata melalui *gesture recognition task* yang disediakan untuk mengenali gerakan statis serta juga menggunakan *holistic landmark* yang mengambil bagian *hand landmark* untuk mengenali gerakan tangan dan menghasilkan *key value* untuk pengumpulan dataset yang kemudian akan digunakan untuk melatih model machine learning melalui LSTM arsitektur.

BAB III

METODE PENELITIAN

3.1 Rancangan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dan kualitatif, yang digunakan untuk menganalisis data *gesture* bahasa isyarat secara numerik dan statistik. Pendekatan ini dipilih karena penelitian ini bertujuan untuk mengembangkan model machine learning yang dapat mengenali dan memahami bahasa isyarat dengan akurasi tinggi. Model machine learning membutuhkan data yang terstruktur dan terukur agar dapat dilatih dan dievaluasi dengan baik. Pendekatan kuantitatif ini juga akan mendukung penelitian ini dimana memerlukan dokumen yang berisi data-data gambar ataupun video yang dibutuhkan untuk dataset akan diubah ke bentuk numerik untuk diolah pada *machine learning*. Kemudian untuk pendekatan kualitatif nya akan memenuhi kebutuhan dari pengembangan, bagaimana memberikan pemahaman yang komprehensif terhadap bahasa isyarat dan memperkuat validitas penelitian ini. Untuk metode yang digunakan adalah studi kasus dimana melakukan penelitian yang difokuskan untuk menggali dan mengumpulkan data yang lebih dalam terhadap objek yang diteliti dan mengembangkan suatu aplikasi untuk dapat menjawab permasalahan yang sedang terjadi.

Teknik pengumpulan data yang digunakan pada penelitian ini memiliki beberapa jenis metode sehingga mampu meningkatkan kinerja model yang dibuat. Data yang digunakan dalam penelitian ini terdiri dari data sekunder dan data primer. Data sekunder dikumpulkan dari pihak ketiga atau penyedia dataset yang berisi database gestur bahasa isyarat, dan pedoman terkait, seperti yang dapat ditemukan melalui platform seperti Kaggle atau data.gov. Data ini digunakan untuk melatih model machine learning. Data primer dikumpulkan dengan merekam *gesture* bahasa isyarat secara langsung untuk menambah variasi data dan mengevaluasi model machine learning.

Untuk teknik pengumpulan data sekunder, data dapat ditelusuri melalui browser pihak ketiga yang menyediakan dokumen dataset gesture seperti Kaggle, data.gov, dan lainnya. Sedangkan untuk pengumpulan data primer akan dilakukan dengan:

1. **Metode:** Pengumpulan data dilakukan dengan menggunakan kamera untuk merekam gerakan tangan yang sesuai dengan panduan SIBI.
2. **Instrumen:** Kamera dan program yang memiliki algoritma untuk mengolah data, mengidentifikasi, serta mencatat gerakan tangan.
3. **Prosedur:** Partisipan akan diminta untuk melakukan berbagai gerakan tangan sesuai dengan kamus SIBI. Data gerakan tangan akan direkam dan disimpan dalam format yang dapat diproses oleh MediaPipe.
4. **Tuli total:** yaitu mereka yang sudah sama sekali tidak memiliki pendengaran sehingga tidak dapat digunakan untuk menyimak atau mempersepsi dan mengembangkan bicara.

3.2 Tahapan Penelitian

Pelaksanaan penelitian ini dilakukan melalui berbagai langkah secara bertahap. Langkah atau tahapan penelitian yang diterapkan disusun sedemikian rupa untuk keefektifan dan kelancaran penelitian. Tahapan pada penelitian ini meliputi:

1. **Studi Literatur:** sebelum memulai penelitian, mengumpulkan referensi dan informasi terkait hal-hal yang menjadi permasalahan yang ingin diselesaikan seperti implementasi machine learning terhadap penerjemahan bahasa isyarat di Indonesia. Penelusuran pustaka, jurnal, artikel, dan penelitian sebelumnya yang relevan dengan topik penelitian. Mengidentifikasi metode, teknik, dan teknologi terbaru yang dapat digunakan dalam penelitian ini.
2. **Identifikasi Kebutuhan:** Mengidentifikasi apa saja yang akan dibutuhkan dalam proses pengerjaan penelitian dan batasan-batasannya. Menyusun daftar kebutuhan penelitian, termasuk perangkat keras (kamera, komputer), perangkat lunak (library machine learning, tools pengolahan

data), dan sumber daya lainnya. Menentukan batasan penelitian seperti jenis bahasa isyarat yang digunakan (SIBI), jenis data yang akan dikumpulkan, dan ruang lingkup penelitian.

3. **Pengumpulan Data:** mengumpulkan data yang dibutuhkan untuk melatih dan menguji model machine learning. Pengumpulan data dilakukan dengan menggunakan kamera untuk merekam gerakan tangan yang digunakan sesuai dalam panduan SIBI. Data yang dikumpulkan mencakup berbagai variasi gerakan tangan untuk memastikan model dapat mengenali dan memahami bahasa isyarat dengan akurasi tinggi.
4. **Preprocessing Data:** memproses data mentah menjadi format yang siap digunakan untuk pelatihan model. mengonversi data menjadi format numerik yang dapat diolah oleh algoritma machine learning.
5. **Pengembangan Mode:** mengembangkan model machine learning yang dapat mengenali dan memahami bahasa isyarat. Menggunakan library MediaPipe Solution dan melakukan kustomisasi sesuai kebutuhan penelitian
6. **Pelatihan Model:** melatih model dengan dataset yang telah diproses. Melakukan training model dengan menggunakan dataset yang ada dengan menggunakan teknik-teknik optimasi dan percobaan inputan parameter untuk meningkatkan akurasi model.
7. **Pengujian dan Evaluasi:** menguji model dengan data uji dan mengevaluasi akurasi model serta loss pada validasi data uji.
8. **Pengembangan Mobile app:** mengembangkan aplikasi mobile yang nantinya akan menjadi platform untuk implementasi model. Merancang dan mengembangkan aplikasi mobile yang dapat mengintegrasikan model machine learning. Memastikan aplikasi user-friendly dan dapat diakses oleh pengguna dengan mudah. Mengimplementasikan fitur-fitur yang mendukung penerjemahan bahasa isyarat secara real-time.
9. **Uji Coba:** menguji aplikasi pada berbagai kondisi penggunaan nyata untuk mengukur keefektifannya. Mengumpulkan umpan balik dari pengguna untuk mengetahui pengalaman mereka dalam menggunakan

aplikasi. Mengidentifikasi area yang perlu perbaikan berdasarkan umpan balik dan hasil uji coba.

10. **Penyusunan Laporan:** menyusun laporan yang mendokumentasikan setiap tahapan penelitian, metode yang digunakan, hasil yang diperoleh, serta analisis dan kesimpulan. Menyajikan laporan dalam format yang sesuai dengan standar akademik.

3.3 Objek Penelitian

Objek penelitian ini adalah gerakan tangan, dimana ini dipilih karena berdasarkan relevansinya dengan tujuan dari penelitian ini yakni membuat suatu penerjemah bahasa isyarat melalui pengenalan gesture tangan. Penelitian ini akan mengamati dan mengidentifikasi bagaimana gerakan tangan menentukan makna yang ingin disampaikan oleh orang tersebut.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Pada tahap ini, penting untuk mengidentifikasi dan memahami berbagai kebutuhan yang harus dipenuhi oleh sistem agar dapat berfungsi sesuai dengan tujuan yang diinginkan. Analisis kebutuhan ini mencakup kebutuhan fungsional dan non-fungsional yang akan memandu pengembangan aplikasi penerjemah bahasa isyarat berbasis MediaPipe.

4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang harus dipenuhi oleh sistem agar dapat berfungsi dengan baik. Dalam pengembangan aplikasi penerjemah bahasa isyarat berbasis MediaPipe, kebutuhan fungsional yang diidentifikasi adalah sebagai berikut:

1. Deteksi Gerakan Tangan

- Sistem harus dapat mendeteksi gerakan tangan secara real-time menggunakan kamera.
- Sistem harus mampu mengidentifikasi posisi dan orientasi tangan untuk setiap frame video yang ditangkap.

2. Pengenalan Gestur Bahasa Isyarat

- Sistem harus mampu mengenali dan menginterpretasi berbagai gestur bahasa isyarat yang telah diprogram.
- Sistem harus memiliki akurasi yang tinggi dalam pengenalan gestur untuk memastikan hasil terjemahan yang benar.

3. Terjemahan Bahasa Isyarat ke Teks

- Sistem harus dapat menerjemahkan gestur bahasa isyarat yang terdeteksi menjadi teks yang dapat dibaca oleh pengguna.
- Terjemahan harus dilakukan secara real-time untuk mendukung komunikasi yang lancar.

4. Antarmuka Pengguna yang Interaktif

- Aplikasi harus menyediakan antarmuka yang mudah digunakan oleh pengguna, termasuk pengguna tunarungu.
- Antarmuka harus memungkinkan pengguna untuk memulai dan menghentikan deteksi gestur, serta melihat hasil terjemahan secara langsung.

4.1.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah karakteristik kualitas yang harus dimiliki oleh sistem agar dapat berfungsi dengan optimal. Kebutuhan non-fungsional yang diidentifikasi untuk aplikasi ini adalah sebagai berikut:

1. Kinerja

- Sistem harus mampu memproses video dan mengenali gestur dengan kecepatan yang cukup sehingga tidak ada keterlambatan yang signifikan dalam terjemahan.
- Sistem harus mampu menangani variasi dalam kualitas video dan kondisi pencahayaan yang berbeda.

2. Keandalan

- Sistem harus memiliki tingkat keandalan yang tinggi dan mampu beroperasi secara terus-menerus tanpa mengalami crash atau kesalahan yang signifikan.
- Sistem harus memiliki mekanisme untuk menangani kesalahan dalam pengenalan gestur dan memberikan umpan balik yang tepat kepada pengguna.

3. Portabilitas

- Aplikasi harus dapat dijalankan di berbagai perangkat, termasuk smartphone dan komputer dengan berbagai sistem operasi.
- Sistem harus dirancang untuk dapat diperluas dan ditingkatkan di masa mendatang.

4. Keamanan dan Privasi

- Sistem harus memastikan bahwa data video yang ditangkap dan diterjemahkan tidak disalahgunakan dan dilindungi dari akses yang tidak sah.
- Sistem harus mematuhi standar keamanan dan privasi data yang berlaku.

4.1.3 Kebutuhan Pengguna

Kebutuhan pengguna mencakup harapan dan kebutuhan spesifik dari pengguna aplikasi ini. Pengguna utama dari aplikasi ini adalah individu tunarungu dan orang yang berinteraksi dengan mereka. Kebutuhan pengguna yang diidentifikasi adalah sebagai berikut:

1. Kemudahan Penggunaan

- Pengguna memerlukan antarmuka yang intuitif dan mudah dipahami tanpa memerlukan pelatihan khusus.
- Pengguna harus dapat dengan mudah mengakses dan menggunakan fitur-fitur utama aplikasi.

2. Akurasi dan Keandalan

- Pengguna mengharapkan terjemahan yang akurat dan dapat diandalkan untuk memastikan komunikasi yang efektif.
- Pengguna memerlukan umpan balik yang jelas dan cepat mengenai hasil terjemahan.

3. Fleksibilitas

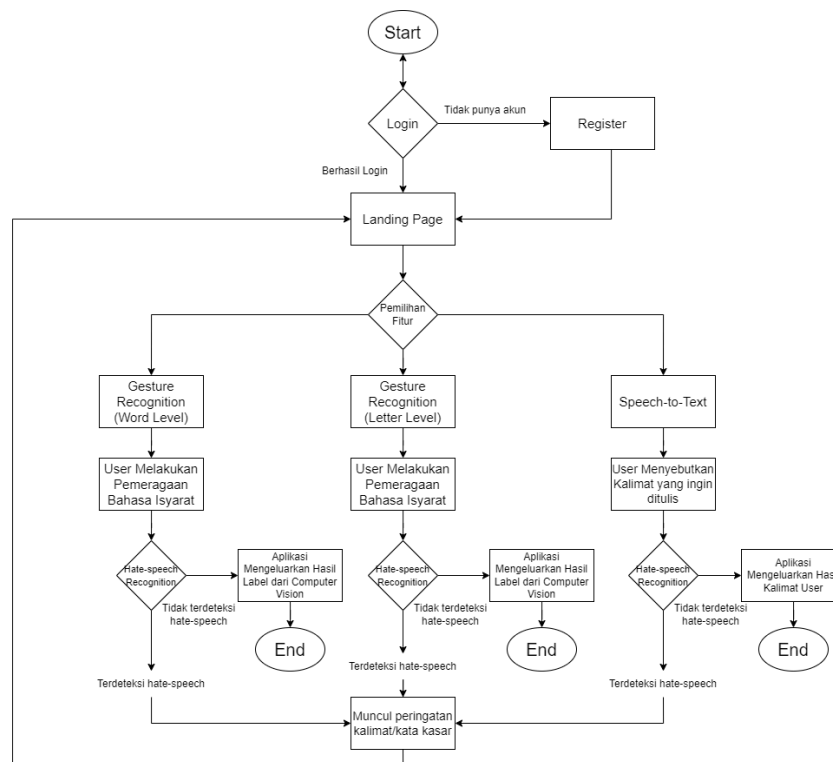
- Pengguna membutuhkan aplikasi yang dapat digunakan dalam berbagai situasi dan lingkungan, baik di dalam ruangan maupun di luar ruangan.
- Pengguna menginginkan aplikasi yang dapat menyesuaikan dengan kebutuhan individu mereka, termasuk penyesuaian ukuran teks dan kecepatan terjemahan.

Dengan memahami dan memenuhi kebutuhan-kebutuhan ini, diharapkan aplikasi penerjemah bahasa isyarat berbasis MediaPipe dapat meningkatkan aksesibilitas komunikasi bagi tunarungu dan memberikan manfaat yang signifikan bagi penggunaanya.

4.2 Perancangan Aplikasi

Perancangan aplikasi adalah tahap penting yang menentukan bagaimana sistem akan diimplementasikan berdasarkan analisis kebutuhan yang telah dilakukan. Dalam tahap ini, berbagai aspek teknis dan desain aplikasi direncanakan untuk memastikan bahwa aplikasi penerjemah bahasa isyarat berbasis MediaPipe dapat berfungsi dengan baik dan memenuhi kebutuhan pengguna. Bagian ini akan membahas arsitektur sistem, desain antarmuka pengguna, serta alur kerja aplikasi secara rinci.

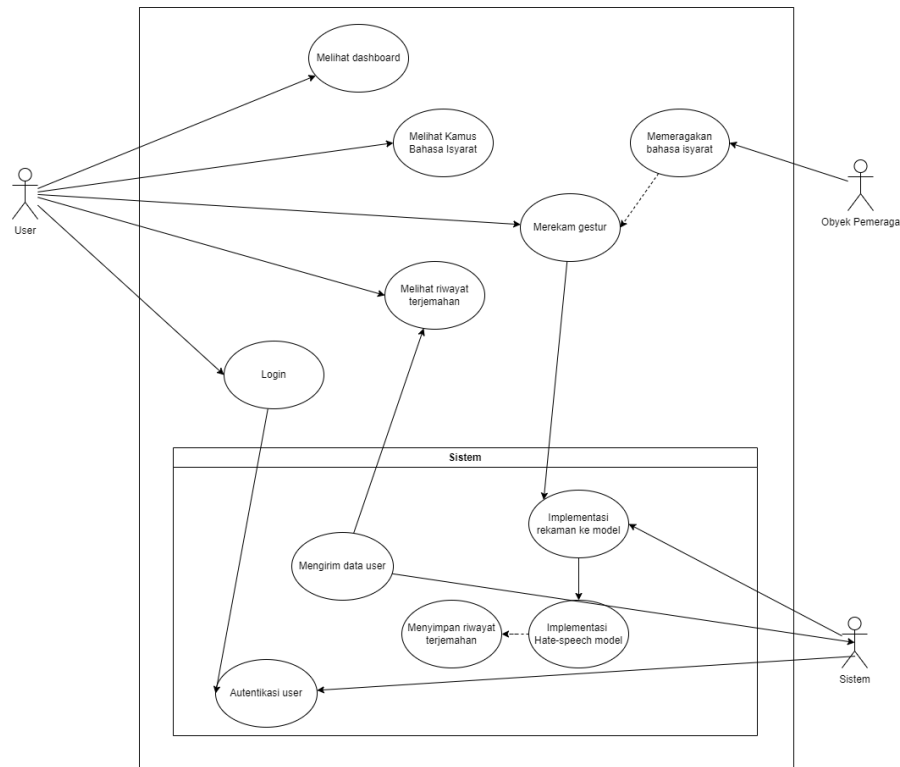
4.2.1 Flowchart



Gambar 4.1 Flowchart Diagram

Pada Gambar 4.1 menjelaskan tentang alur kerja aplikasi penerjemah bahasa isyarat yang memiliki beberapa fitur, yaitu *Gesture Recognition (word level)*, *Gesture Recognition (letter level)*, dan *Speech-to-Text*. Sebelum menggunakan fitur yang terdapat pada aplikais ini, pengguna diharuskan untuk melakukan login dan register jika belum memiliki akun. Setelah berhasil login, user bebas mencoba tiga fitur utama pada aplikasi penerjemah bahasa isyarat ini. Selain itu, pada penggunaan aplikasi ini juga terdapat *Hate Speech Detection* atau Pendeteksi Kalimat Kasar yang berfungsi untuk membatasi pengguna dalam menggunakan aplikasi ini untuk melakukan tindakan kejahatan pada pengguna lainnya. Saat pengguna menggunakan salah satu fiturnya dan hasil label dari computer vision yang diperagakan atau dikatakan pengguna terbukti terdeteksi kalimat kasar, user akan mendapatkan peringatan berbentuk modal dan dipaksa untuk keluar fitur dan kembali ke halaman awal.

4.2.2 Use Case Diagram



Gambar 4.2 Use Case Diagram

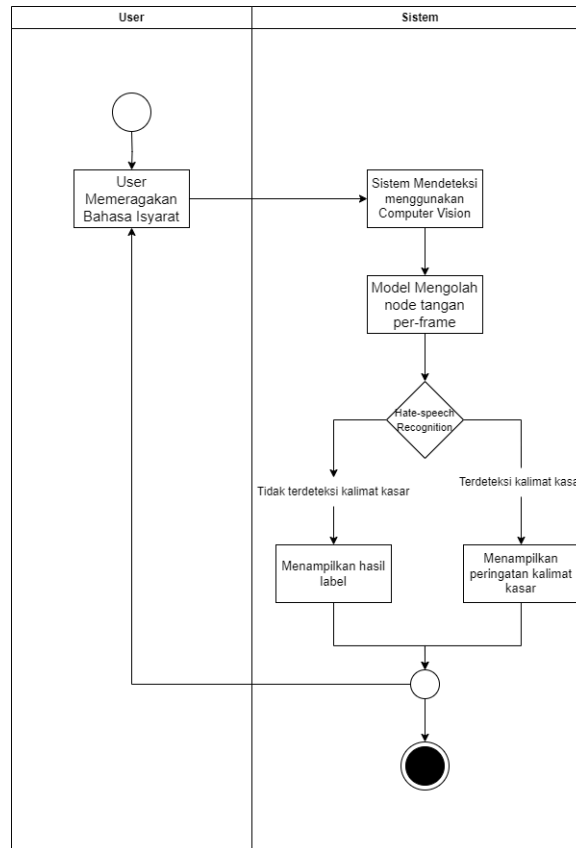
Pada fungsional aplikasi ini, terdapat dua aktor yang saling berkaitan untuk menerjemahkan bahasa isyarat. Aktor pertama adalah orang yang kurang paham dengan bahasa isyarat yang akan memegang aplikasi mobile ini dan mengarahkannya kepada aktor kedua. Aktor kedua adalah orang yang memeragakan bahasa isyarat yang akan diterjemahkan sistem kepada aktor pertama. Setiap aksi yang diperagakan aktor kedua akan selalu dicek oleh sistem menggunakan model hate-speech berbasis NLP untuk mendeteksi kalimat kasar yang bisa saja terjadi jika aktor kedua memeragakan hal serupa.

4.2.3 Activity Diagram

Setelah memahami perancangan aplikasi secara keseluruhan, penting untuk menggambarkan alur aktivitas yang terjadi dalam sistem. Activity Diagram membantu

dalam memvisualisasikan proses dan interaksi yang terjadi di dalam aplikasi, memberikan gambaran yang lebih jelas mengenai bagaimana sistem bekerja. Pada bagian ini, akan dijelaskan Activity Diagram yang meliputi berbagai fitur utama dari aplikasi.

4.2.3.1 Activity Diagram fitur Word Level Gesture Recognition

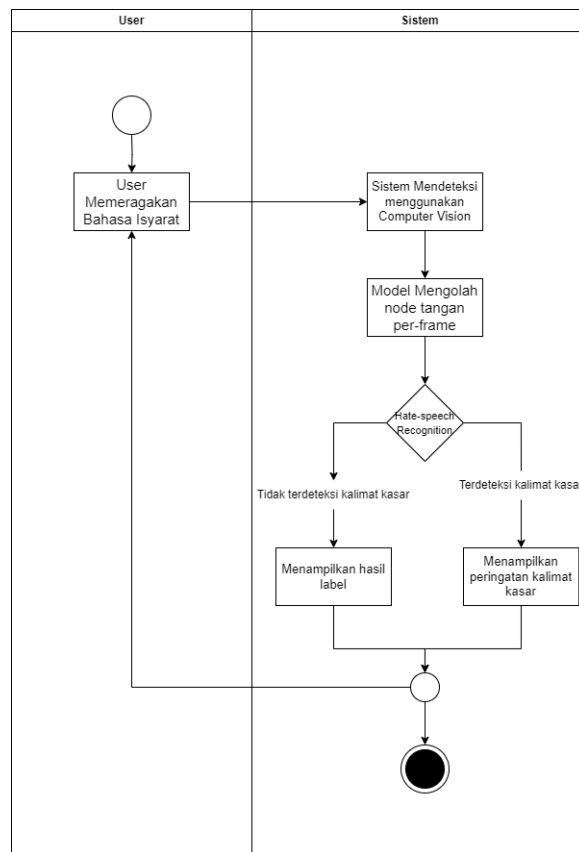


Gambar 4.3 Activity Diagram *Gesture Recognition* (Word Level)

Pada gambar 4.3 diatas merupakan alur aktivitas yang dilakukan oleh pengguna dan sistem pada salah satu fitur utama pada pengembangan aplikasi ini yaitu *Word Level Gesture Recognition*. Pada fitur ini, user cukup melakukan gerakan bahasa isyarat sesuai dengan Kamus SIBI yang bisa dilihat pada fitur kamus. Setelah itu, sistem akan lanjut melakukan deteksi node pada gerakan tangan pengguna setiap 30 fps. Setelah sistem berhasil menentukan label output yang dihasilkan dari gerakan pengguna, sistem akan lanjut mendeteksi konteks kalimat yang sudah diperagakan

pengguna. Jika sistem mendeteksi ujaran kebencian atau kalimat kasar pada output gerak pengguna, sistem akan mengeluarkan peringatan dan dipaksa untuk keluar fitur, sedangkan jika sistem tidak mendeteksi ujaran kebencian atau kalimat kasar pada output gerakan pengguna, sistem akan terus berjalan hingga pengguna memberhentikanannya.

4.2.3.2 Activity Diagram fitur Letter Level Gesture Recognition

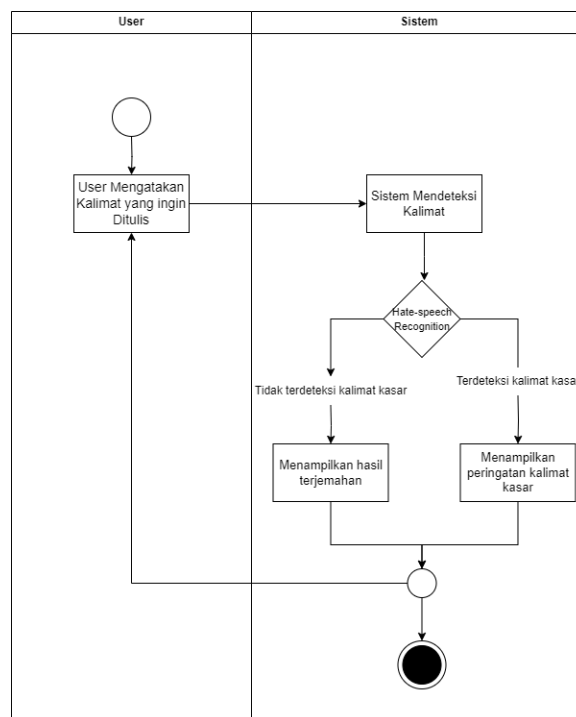


Gambar 4.4 Activity Diagram *Gesture Recognition* (Letter Level)

Pada gambar 4.4 diatas merupakan alur aktivitas yang dilakukan oleh pengguna dan sistem pada salah satu fitur utama pada pengembangan aplikasi ini yaitu *Letter Level Gesture Recognition*. Alur aktivitas antara sistem dan pengguna ini tidak jauh berbeda dengan fitur pertama, akan tetapi pengguna menggunakan gerakan bahasa isyarat perhuruf yang akan membentuk kata dan kalimat sebagai output. Pada fitur ini, pengguna

cukup melakukan gerakan bahasa isyarat sesuai dengan Kamus SIBI yang bisa dilihat pada fitur kamus. Setelah itu, sistem akan lanjut melakukan deteksi node pada gerakan tangan pengguna setiap 30 fps. Setelah sistem berhasil menentukan label output yang dihasilkan dari gerakan pengguna, sistem akan lanjut mendeteksi konteks kalimat yang sudah diperagakan pengguna. Jika sistem mendeteksi ujaran kebencian atau kalimat kasar pada output gerak pengguna, sistem akan mengeluarkan peringatan dan dipaksa untuk keluar fitur, sedangkan jika sistem tidak mendeteksi ujaran kebencian atau kalimat kasar pada output gerakan pengguna, sistem akan terus berjalan hingga pengguna memberhentikannya.

4.2.3.3 Activity Diagram fitur Speech-to-Text



Gambar 4.5 Activity Diagram *Speech-to-Text*

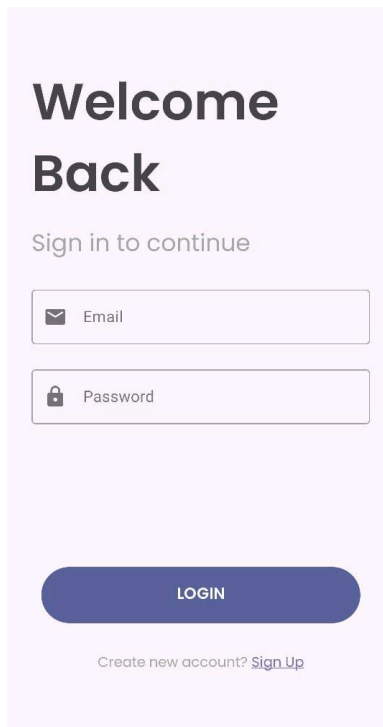
Pada gambar 4.5 diatas merupakan merupakan alur aktivitas yang dilakukan oleh pengguna dan sistem pada salah satu fitur utama pada pengembangan aplikasi ini yaitu *Speech-to-Text*. Pada fitur ini, pengguna

cukup mengatakan kalimat apapun yang ingin ditulis di dalam whiteboard aplikasi. Setelah itu sistem akan mendeteksi ujaran kebencian atau kalimat kasar dari ucapan pengguna yang sudah diimplementasi pada whiteboard aplikasi. Jika sistem mendeteksi ujaran kebencian atau kalimat kasar pada output ucapan pengguna, sistem akan mengeluarkan peringatan dan dipaksa untuk keluar fitur, sedangkan jika sistem tidak mendeteksi ujaran kebencian atau kalimat kasar pada output ucapan pengguna, sistem akan terus berjalan hingga pengguna memberhENTikannya.

4.3 Implementasi Sistem

Dengan perancangan aplikasi yang telah dijelaskan, langkah selanjutnya adalah implementasi sistem untuk merealisasikan desain tersebut. Bagian ini akan menjelaskan bagaimana setiap komponen yang telah dirancang diimplementasikan secara teknis, dengan fokus pada pengembangan antarmuka pengguna (UI) untuk aplikasi mobile. Implementasi sistem ini mencakup detail teknis mengenai integrasi MediaPipe, pemrosesan data, dan penyusunan antarmuka yang user-friendly guna memastikan aplikasi dapat berfungsi secara optimal dan memberikan pengalaman pengguna yang baik. Berikut beberapa bagian halaman pada implementasi aplikasi mobile beserta sedikit fungsionalitas pada setiap fitur:

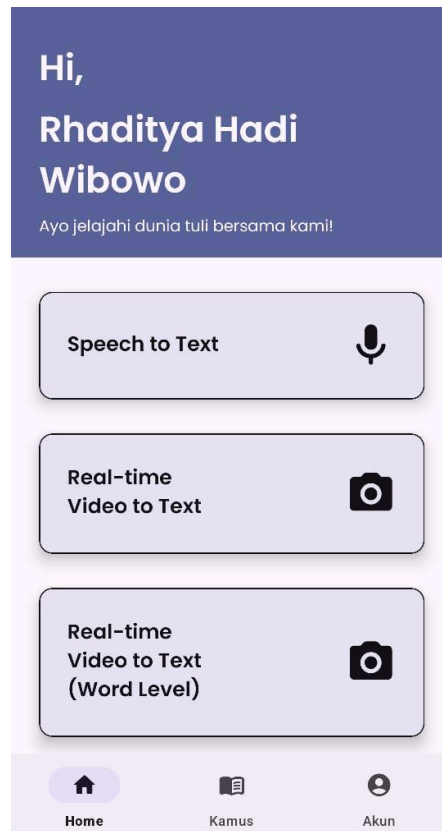
4.3.1 Halaman Login



Gambar 4.3.1 Tampilan UI Halaman Login

Sebelum pengguna memasuki aplikasi, pengguna diwajibkan untuk melakukan login atau register jika belum memiliki akun. Kami menggunakan tools seperti firebase authentication untuk menampung login session pengguna dan firestore untuk menampung data pengguna.

4.3.2 Halaman Utama



Gambar 4.3.2 Tampilan UI Halaman Utama

Setelah pengguna melakukan login, pengguna akan langsung diarahkan ke halaman utama dari aplikasi ini, yaitu halaman yang memiliki segala akses kepada fitur-fitur utama aplikasi gestura.

4.3.3 *Letter Level Gesture Recognition*

Letter Level Gesture Recognition, pengenalan gerakan tangan untuk mengenali huruf-huruf secara individual. Sistem ini dirancang untuk mendeteksi gerakan tangan pengguna yang merepresentasikan huruf-huruf tertentu, yang kemudian diterjemahkan menjadi input teks.

4.3.3.1 Load Dataset

Sebelum memulai training memastikan dataset sudah disediakan terlebih dahulu dan dataset yang dipakai disini merupakan kombinasi SIBI dan ASL dataset, dengan alasan bahwa untuk keakuratan dan peningkatan training pada model. Dimana pada dataset SIBI untuk kasus seperti huruf M dan N memiliki gesture yang hampir mirip sehingga akan menyulitkan pelatihan pengenalan pada model. Berikut adalah contoh sample dataset huruf.



Gambar 4.3.3.1.1 Examples dataset

Jumlah label pada *Letter Level Gesture* ini adalah 39 label atau kelas, dimana ini adalah gabungan dari alphabet dan angka serta penambahan kelas 'del' yang akan berfungsi untuk menghapus huruf yang sudah disimpan sebagai kata, 'space' berfungsi untuk menambah spasi pada kumpulan huruf yang membentuk kata hingga kalimat dan kelas 'none' sebagai syarat dari pengaplikasian menggunakan mediapipe. Dataset akan dibagi kedalam 3 variabel, yaitu *train_data*, *validation_data* dan *test_data* dengan kondisi pertama mengambil *train_data* hingga 80% dari jumlah

dataset yang kemudian sisanya dibagi 2 untuk *validation_data* dan *test_data*. Dengan menggunakan metode *Dataset.from_folder API* dari *gesture_recognizer*, struktur dari dataset harus dalam format *<root_dir>/<gesture_name>/*.jpg* dan juga harus ada *label* atau class yang bernama *none* yang berisi gambar tidak sesuai.

```
data = gesture_recognizer.Dataset.from_folder(
    dataset_path,
    hparams=gesture_recognizer.HandDataPreprocessingParams()
)
train_data, rest_data = data.split(0.8)
validation_data, test_data = rest_data.split(0.5)
```

Gambar 4.3.3.1.2 Split Data

Hasilnya akan mendeteksi landmark tangan dari gambar. Semua gambar tanpa tangan yang terdeteksi akan dihilangkan dari set data pada setiap classnya. Set data yang dihasilkan akan berisi posisi landmark tangan yang diekstrak dari setiap gambar.

4.3.3.2 Model

```
Model: "model_4"
=====
```

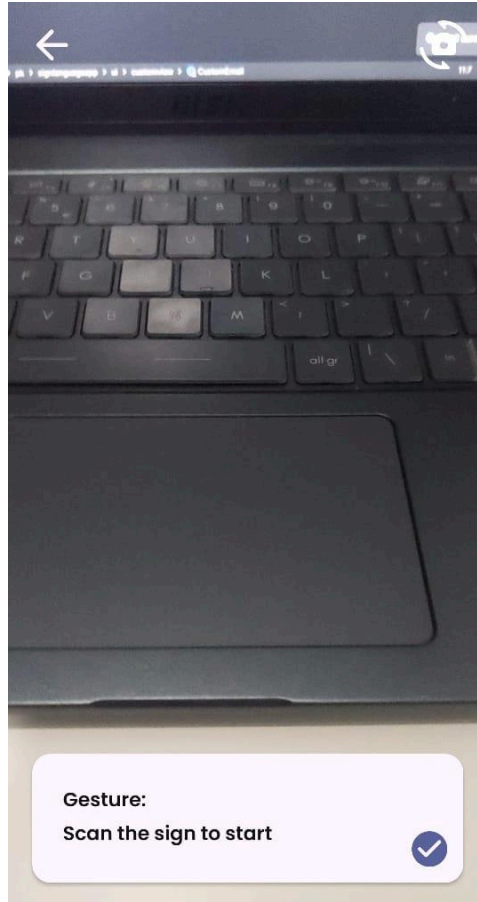
Layer (type)	Output Shape	Param #
hand_embedding (InputLayer)	[(None, 128)]	0
batch_normalization_4 (Batch Normalization)	(None, 128)	512
re_lu_4 (ReLU)	(None, 128)	0
dropout_4 (Dropout)	(None, 128)	0
custom_gesture_recognizer_out (Dense)	(None, 39)	5031

```
=====
Total params: 5543 (21.65 KB)
Trainable params: 5287 (20.65 KB)
Non-trainable params: 256 (1.00 KB)
```

Gambar 4.3.3.2.1 Model summary()

Dengan menggunakan fungsi `summary()` pada model, kita dapat memperoleh informasi rinci mengenai struktur dan parameter yang terdapat dalam model. Fungsi ini memberikan gambaran tentang jumlah layer, bentuk output dari setiap layer, serta jumlah parameter yang dapat dilatih. Informasi ini berfungsi dan untuk mengidentifikasi area yang dapat diubah guna meningkatkan akurasi model. Pada gambar diatas mengambil hasil dari model yang menggunakan API mediapipe yang memiliki output hand landmark detection sebagai inputan dan dilanjutkan untuk pelatihan deteksi sesuai class dataset dengan metode neural network yang menggunakan *batch_normalization* dimana teknik ini meningkatkan pelatihan dengan melakukan normalisasi terhadap inputan aktivasi sehingga lebih stabil dan juga menggunakan *relu* activation serta dropout dengan Dense layer yang memiliki output 39 class.

Kemudian juga melakukan fine-tuning pada hyperparameter merupakan langkah penting dalam mengoptimalkan performa model. Dalam konteks penggunaan API Mediapipe fine-tuning model dapat menggunakan `GestureRecognizerOptions`, yang memiliki dua parameter opsional untuk `ModelOptions` dan `HParams`. `ModelOptions` untuk menyesuaikan parameter yang terkait dengan model itu sendiri, dan kelas `HParams` untuk menyesuaikan parameter lain yang terkait dengan pelatihan dan penyimpanan model.



Gambar 4.11 Hasil Implementasi Model

Implementasi model *letter-level recognition* pada aplikasi menggunakan *library* mediapipe. Pada *letter-level*, abjad yang diprediksi dapat dibuat menjadi sebuah kata bahkan kalimat. Abjad yang diprediksi dapat dimasukan kedalam sebuah string dalam waktu 2 detik berdasarkan konsistensi dalam jangka waktu tersebut. Fitur ini juga dilengkapi *hate speech detection* yang memunculkan alert untuk membatasi pengguna dalam berbuat hal yang tidak etis.

4.3.4 *Word Level Gesture Recognition*

4.3.4.1. Data Extraction dan Feature Extraction

Dataset yang kami miliki merupakan dataset primer yang dikumpulkan secara manual dengan menggunakan Mediapipe Hand Landmarker. Setiap dataset ini mengandung ekstraksi 30 baris koordinat dari 21 titik pada tangan yang dikomputasi oleh Mediapipe, dan kami melakukan ekstraksi tiap 30 frame kamera. Label yang kami miliki berjumlah 217, yang mencakup berbagai gerakan tangan. Setiap label memiliki setidaknya 60 data latih, dimana tiap data latih terdiri dari 30 baris koordinat dari 21 titik koordinat hand landmark. Mediapipe Hand Landmarker dikonfigurasi untuk mengenali dua tangan secara bersamaan, sehingga input shape yang kami miliki adalah $(30, 63 * 2)$, yang merepresentasikan 30 frame, 63 titik koordinat per tangan, dan 2 tangan. Dalam keseluruhan dataset, kami memiliki total 11,240 contoh data. Proses ekstraksi fitur ini penting untuk menangkap dinamika gerakan tangan yang merupakan inti dari penerjemahan bahasa isyarat.

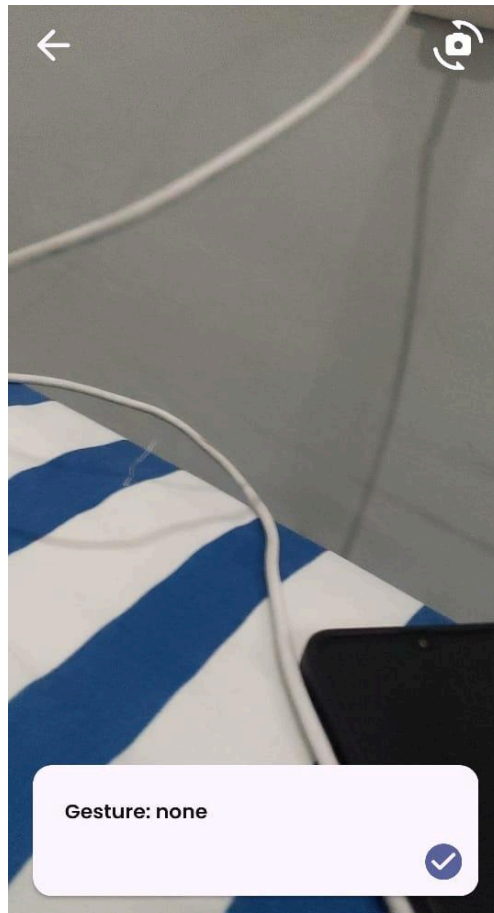
4.3.4.2. Data Preparation

Setelah ekstraksi fitur, kami melakukan proses flatten pada dataset kami. Awalnya, dataset memiliki dimensi $(30, 21, 3)$, yang merepresentasikan 30 frame, 21 titik, dan 3 koordinat (x, y, z) . Proses flatten mengubah dimensi ini menjadi $(30, 63)$, dengan menggabungkan titik koordinat untuk setiap frame menjadi satu vektor. Proses flatten ini dilakukan untuk mempermudah pengolahan data dalam tahap pelatihan model, terutama dalam arsitektur model yang kami gunakan. Selain itu, dengan mengurangi kompleksitas dimensi, kami dapat meningkatkan

efisiensi komputasi dan mempercepat proses pelatihan. Kami membagi dataset dengan rasio 70:30 untuk train dan validation, yang bertujuan untuk memastikan bahwa model kami dapat dilatih dengan cukup data sambil tetap mempertahankan set data yang cukup besar untuk validasi guna mengevaluasi performa model.

4.3.4.3. Model Preparation

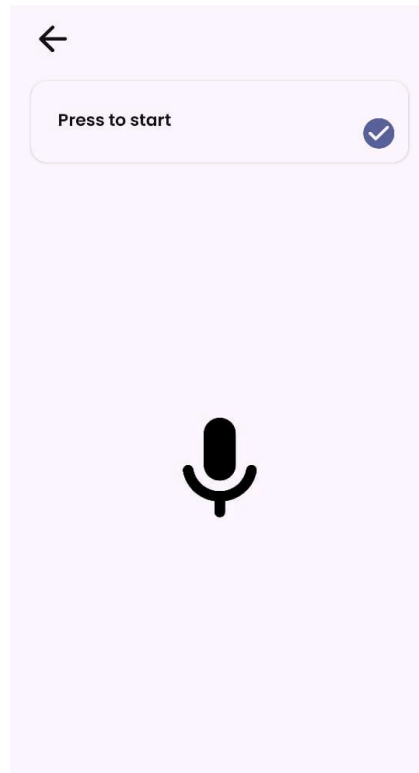
Arsitektur yang kami gunakan adalah Long Short-Term Memory (LSTM), yang dikenal efektif dalam menangani data berurutan seperti video atau sinyal waktu. LSTM dapat mempertahankan informasi dari frame sebelumnya dan memanfaatkannya untuk memprediksi frame selanjutnya, yang sangat penting dalam mengenali pola gerakan tangan. Selain LSTM, kami menambahkan beberapa layer tambahan seperti Dropout untuk mencegah overfitting dan Batch Normalization untuk mempercepat proses pelatihan dan meningkatkan stabilitas model. Untuk proses finetuning, kami menetapkan 200 epoch, learning rate sebesar $1e-3$, dan menggunakan optimizer Adam yang dikenal karena keefektifannya dalam menemukan solusi optimal dengan cepat.



Gambar 4.3.4.3.3 Hasil Implementasi Model di Aplikasi Mobile

Implementasi model *word-level recognition* pada aplikasi menggunakan *library* mediapipe. Fitur ini juga dilengkapi *hate speech detection* yang memunculkan alert untuk membatasi pengguna dalam berbuat hal yang tidak etis.

4.3.5 *Speech-to-Text Recognition*



Gambar 4.3.5 Tampilan UI Fitur Speech-to-Text

Fitur ini dibangun menggunakan *package* yang disediakan oleh Google dalam pengembangan android. Pada kasus kali ini, bahasa yang didukung hanyalah bahasa Indonesia. Selain itu, fitur ini juga dilengkapi oleh fitur Hate Speech Detection.

4.3.6 Hate Speech Detection

4.3.6.1 Data Preprocessing

	text	labels
0	knp gak berani bersumpah dan bertaruh kmlo pe...	1
1	haha somad somad muka dekil otak kok malu ngak...	1
2	hahaha kaum sableng berita begini mrk buta tuli	1
3	hahaha makin stress aja umat sableng dlu raja ...	1
4	hidup psi partai sampah indonesia	1

Gambar 4.3.6.1.1 Hasil *Text Cleaning*

Pada tahap awal preprocessing data, kami menggunakan metode text cleaning untuk membersihkan data dari segala simbol ataupun kata typo yang sering muncul dalam dataset sentimen analisis. Contoh simbol-simbol yang dibersihkan yaitu @, #, kata non-alphabetical, dan angka. Selain itu, kami juga menggunakan metode stop word untuk memisahkan kalimat per kalimat untuk menentukan konteks. Pada gambar 4.3.6.1.1 diatas, terlihat bahwa beberapa sampel data yang sudah dibersihkan dari segala simbol yang dituliskan. Pada gambar tersebut juga terlihat hasil kalimat dari metode stop word untuk memudahkan model menentukan konteks.

	text	labels	stemmed_text
0	knp gak berani bersumpah dan bertaruh kmlo pe...	1	knp tidak berani sumpah dan taruh kmlo pecund...
1	haha somad somad muka dekil otak kok malu ngak...	1	haha somad somad muka dekil otak kok malu ngak...
2	hahaha kaum sableng berita begini mrk buta tuli	1	hahaha kaum sableng berita begini mrk buta tuli
3	hahaha makin stress aja umat sableng dlu raja ...	1	hahaha makin stress aja umat sableng dlu raja ...
4	hidup psi partai sampah indonesia	1	hidup psi partai sampah indonesia
...
14301	muka babi ahok tuh mirip serbet lantai	1	muka babi ahok itu mirip serbet lantai
14302	betul bang hancurkan merka bang musnahkan chin...	1	betul bang hancur memerka bang musnah china ba...
14303	sapa bilang ahok anti korupsi klo grombolannya...	1	sapa bilang ahok anti korupsi kalau grombolann...
14304	juga ngimpi sentilin biji babi ahok pcetar pce...	1	juga impi sentil biji babi ahok pcetar pcetar ...
14305	mudahan ketemu sama babi iwan bopeng tps nya a...	1	mudah temu sama babi iwan bopeng tps nya anjin...

Gambar 4.3.6.1.2 Hasil *Text Stemming*

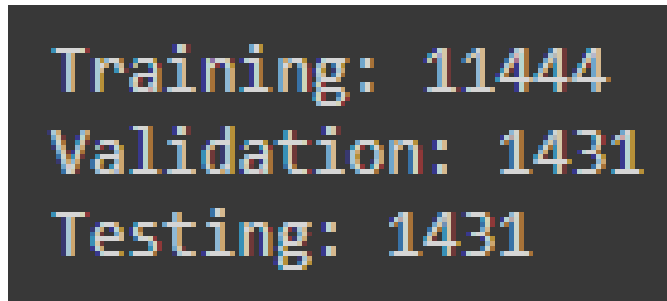
Selanjutnya kami menggunakan metode stemming untuk mengubah kata-kata berimbuhan ataupun kata yang memiliki arti sama menjadi kata aslinya menggunakan library MPStemmer. Hal ini dilakukan untuk mengurangi variasi kata pada dataset sehingga model lebih mudah untuk menghasilkan akurasi dan efisiensi yang baik saat melakukan training. Pada gambar 4/3/6/1/2 diatas, terlihat bahwa kata kata seperti gak, ketemu dan ngimpi diubah menjadi tidak, temu dan impi.

labels		text
0	1	kenapa tidak berani sumpah dan taruh krnlo pec...
1	1	haha somad somad muka dekil otak kok malu meng...
2	1	hahaha kaum sableng berita begini mereka buta ...
3	1	hahaha makin stres saja umat sableng dulu raja...
4	1	hidup psi partai sampah indonesia
...
14301	1	muka babi ahok itu mirip serbet lantai
14302	1	betul bang hancur memerka bang musnah china ba...
14303	1	sapa bilang ahok anti korupsi kalau grombolann...
14304	1	juga impi sentil biji babi ahok pcetar pcetar ...
14305	1	mudah temu sama babi iwan bopeng tps nya anjin...
14306 rows x 2 columns		

Gambar 4.3.6.1.3 Hasil *Converting Slang Word*

Setelah melakukan *text stemming*, kami juga mengurangi *slang word* untuk mengurangi variasi kata pada dataset menggunakan library IndoNLP untuk mengubah *slang word* Indonesia menjadi bahasa yang lebih baku dan dikenali oleh model.

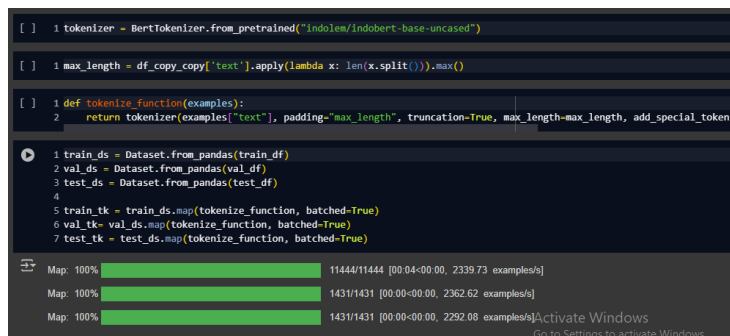
4.3.6.2 Model Preparation



```
Training: 11444
Validation: 1431
Testing: 1431
```

Gambar 4.3.6.2.1 Hasil Split Dataset

Setelah melakukan data preprosesing, kami langsung mengeksekusi model pada tahap model preparation. Tahap awal model preparation adalah data splitting, tahap dimana total 14.000 dataset dibagi 80% training set, 10% validation set, dan 10% testing set untuk mengevaluasi model dengan baik. Hasil pada gambar 4.3.6.2.1 diatas, menunjukkan bahwa sebanyak 11444 data ditetapkan menjadi training set, 1431 data menjadi validation set, dan 1431 data menjadi test set.



```
[ ] 1 tokenizer = BertTokenizer.from_pretrained("indolem/indobert-base-uncased")

[ ] 1 max_length = df_copy["text"].apply(lambda x: len(x.split()), max())

[ ] 1 def tokenize_function(examples):
    2     return tokenizer(examples["text"], padding="max_length", truncation=True, max_length=max_length, add_special_token=True)

1 train_ds = Dataset.from_pandas(train_df)
2 val_ds = Dataset.from_pandas(val_df)
3 test_ds = Dataset.from_pandas(test_df)
4
5 train_tk = train_ds.map(tokenize_function, batched=True)
6 val_tk = val_ds.map(tokenize_function, batched=True)
7 test_tk = test_ds.map(tokenize_function, batched=True)

Map: 100% 11444/11444 [00:04:00.00, 2339.73 examples/s]
Map: 100% 1431/1431 [00:00:00.00, 2362.62 examples/s]
Map: 100% 1431/1431 [00:00:00.00, 2292.08 examples/s]
```

Gambar 4.3.6.2.1 Hasil Tokenizing

Setelah melakukan data splitting, kami melakukan metode Tokenizing untuk memisahkan suatu kalimat menjadi kata kata dengan masing masing nilai tokenisasi. Kami menggunakan pretrained model BERT Tokenizer dari Indobert untuk mempermudah pengerjaan. Pada hasil tokenizing pada

gambar 4.3.6.2.2 diatas, terlihat pada setiap set memiliki sejumlah kurang lebih 2300 nilai tokenisasi.

Epoch	Training loss	Validation loss	Accuracy	F1	Precision	Recall
1	No log	0.756576	0.858141	0.858127	0.858114	0.858141
2	No log	0.951259	0.841370	0.842057	0.855191	0.841370
3	0.060000	0.671665	0.844165	0.843618	0.843882	0.844165
4	0.060000	0.827477	0.849057	0.849295	0.849804	0.849057
5	0.060000	0.938575	0.831586	0.832288	0.846231	0.831586
6	0.042600	0.968102	0.849057	0.849760	0.857784	0.849057
7	0.042600	0.989954	0.854647	0.854589	0.854545	0.854647
8	0.042600	1.000093	0.853249	0.853526	0.854209	0.853249
9	0.024600	1.098125	0.851153	0.851706	0.854596	0.851153
10	0.024600	1.153417	0.846261	0.846137	0.846066	0.846261
11	0.024600	1.156239	0.850454	0.851051	0.854639	0.850454
12	0.017700	1.158673	0.855346	0.855728	0.857025	0.855346
13	0.017700	1.132372	0.855346	0.855018	0.855866	0.855346
14	0.013700	1.146024	0.857442	0.857548	0.857708	0.857442
15	0.013700	1.007975	0.852551	0.853144	0.856837	0.852551
16	0.013700	0.931509	0.857442	0.857202	0.857180	0.857442
17	0.040900	0.967627	0.853948	0.853436	0.853744	0.853948
18	0.040900	0.947725	0.856045	0.855769	0.855769	0.856045
19	0.040900	0.943844	0.849755	0.850093	0.851039	0.849755
20	0.034700	0.943088	0.858141	0.858155	0.858169	0.858141

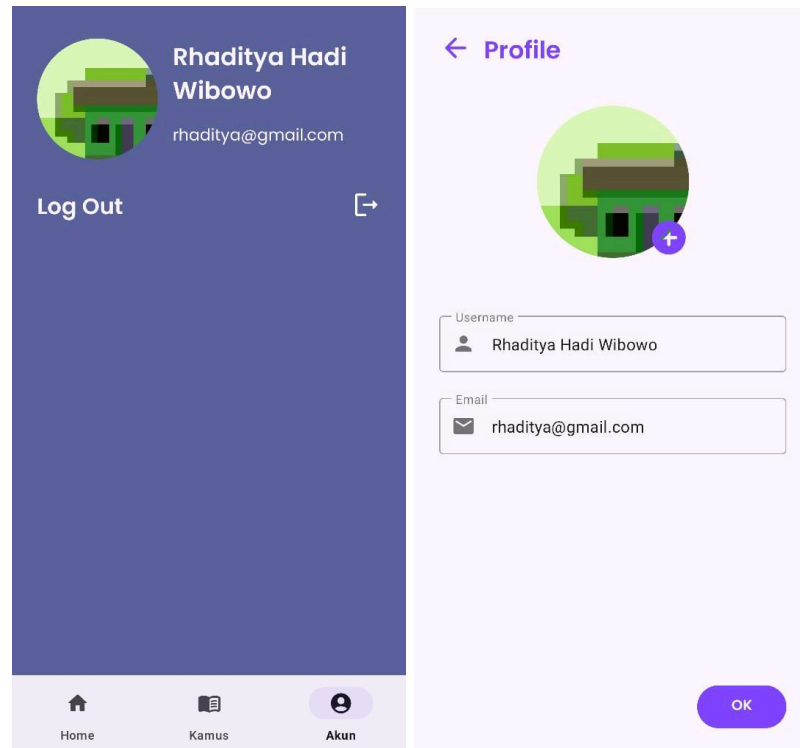
TrainOutput(global_step=3580, training_loss=0.033234544839272954, metrics={'train_runtime': 2138.0064, 'train_samples_per_second': 107.053, 'train_steps_per_second': 1.674, 'total_flos': 5645785420387280.0, 'train_flos': 0.033234544839272954, 'epoch': 20.0})

Gambar 4.21 Hasil Training Model

Selanjutnya, kami menggunakan metode transfer learning dari pretrained model dari IndoBert. Kami menggunakan *task* BertForSequenceClassification dari pretrained IndoBert yang digunakan untuk kebutuhan klasifikasi teks. Pretrained ini nantinya akan dimasukan kedalam parameter training untuk menghasilkan model yang efisien dan optimal. Setelah itu, kami juga menggunakan metode *transfer learning* menggunakan library torch. Kami menambahkan arsitektur GRU (Gated Recurrent Unit) ke dalam pre-trained arsitektur model IndoBert. Seperti LSTM, GRU (Gated Recurrent Unit) bertujuan untuk mengatasi masalah vanishing gradient yang muncul pada jaringan neural berulang standar. GRU menggabungkan forget gate dan input gate menjadi satu update gate dan memiliki reset gate tambahan. GRU semakin populer dan banyak digunakan untuk menyelesaikan masalah NLP. Untuk mengatasi masalah vanishing gradient pada RNN standar, GRU menggunakan update gate dan reset gate. Pada dasarnya, ini adalah dua vektor yang memutuskan informasi apa yang harus diteruskan ke output (Patihullah & Winarko, 2019). Kemudian, kami melakukan beberapa konfigurasi pada

hyperparameter guna dengan tujuan *fine tuning*, seperti learning rate sebesar $1e-5$, jumlah *epoch* sebanyak 20 iterasi, dan dengan batch size berukuran 64 berhubung dataset yang kami miliki cukup besar.

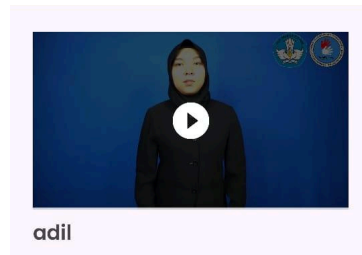
4.3.6 Halaman Profile



Gambar 4.22 Tampilan UI Halaman Profile

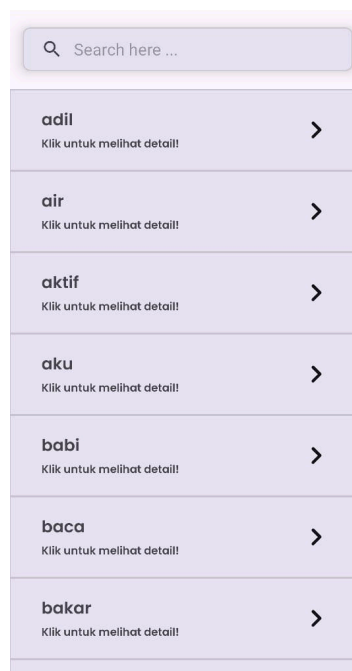
Pada halaman akun, hanya terdapat *card* profil untuk mengakses halaman profil dan *button logout* untuk meng-*logout* akun user. Halaman profil digunakan untuk mengubah foto profil user dan *username* yang akan ditampilkan pada halaman utama dan halaman akun.

4.3.7 Halaman Kamus



Gambar 4.23 Tampilan UI Halaman Kamus

Halaman kamus terdiri dari dua kategori, yaitu abjad dan kata. Kategori abjad meliputi huruf A sampai Z yang dilengkapi gambar gestur tangan. Kategori kata meliputi lebih dari 200 kata yang dilengkapi video untuk memperagakan gestur tersebut.



Gambar 4.24 Tampilan UI Halaman Kamus Berdasarkan Hasil Pencarian

Pada halaman list dari abjad maupun kata dilengkapi fitur *search* untuk memudahkan pengguna dalam mengakses abjad maupun kata yang diinginkan.

4.4 Pengujian

Pengujian dilakukan setelah tahap implementasi yang bertujuan untuk memastikan bahwa aplikasi dapat berjalan dengan baik, tidak terdapat kesalahan sistem, dan sesuai dengan rancangan yang dibuat sebelumnya.

4.4.1 Deskripsi Pengujian

Pada tahap pengujian penelitian ini, kami menggunakan *black box testing* yang dilakukan untuk menguji fungsionalitas aplikasi secara keseluruhan tanpa melihat ke dalam kode sumbernya. Dalam konteks aplikasi penerjemah bahasa isyarat berbasis MediaPipe, blackbox testing menggunakan computer vision bertujuan untuk memastikan bahwa aplikasi dapat mendeteksi dan mengenali gestur bahasa isyarat dengan benar dari input video yang diberikan. Pengujian ini melibatkan pengguna yang melakukan berbagai gestur bahasa isyarat di depan kamera, kemudian aplikasi harus mampu menampilkan terjemahan teks yang sesuai. Fokus dari pengujian ini adalah pada keluaran yang dihasilkan oleh aplikasi berdasarkan masukan gestur tanpa memeriksa proses internalnya.

Disisi lain, kami juga melakukan pengujian model untuk menganalisis kinerja model pada tiap fitur. Pengujian ini menggunakan metode evaluasi seperti akurasi, presisi, recall, dan F1-score untuk mengukur sejauh mana model dapat mengenali gestur bahasa isyarat dengan benar. Kami menguji model ini dengan berbagai dataset yang telah dikumpulkan sebelumnya, yang mencakup beragam gestur dan kondisi pencahayaan yang berbeda. Hasil dari pengujian ini memberikan wawasan tentang kekuatan dan kelemahan model, serta area yang perlu ditingkatkan.

4.4.2 Prosedur Pengujian

4.4.2.1 *Blackbox Testing*

Tabel 4.4.2.1 Prosedur Pengujian *Blackbox Testing*

Item Uji	Detail Pengujian	Pengguna
Login	User login ke akun pribadi	Semua Pengguna
Register	User melakukan register jika belum memiliki akun pribadi	Semua Pengguna
Word Level Gesture Recognition	User memeragakan bahasa isyarat di depan kamera aplikasi	Demonstrator
Letter Level Gesture Recognition	User memeragakan bahasa isyarat di depan kamera aplikasi	Demonstrator
Speech-to-Text	User menyebutkan kalimat yang ingin ditulis	Demonstrator
Hate-speech Recognition	User menggunakan tiga fitur utama dan	Demonstrator

	memeragakan atau menyebutkan kalimat yang kasar atau mengujarkan kebencian	
Kamus Bahasa Isyarat	User melihat referensi bahasa isyarat dari kamus	Semua Pengguna

4.4.2.2 Analisis Model Machine Learning

Tabel 4.4.2.2 Prosedur Pengujian *WhiteBox Testing*

Item Uji	Detail Pengujian
Akurasi	Mengamati akurasi di setiap epoch training model untuk melihat performa model
Training loss	Mengamati loss dari training untuk melihat performa dan membandingkannya dengan validation
Validation loss	Mengamati loss dari validation untuk melihat performa dan membandingkannya dengan training

F1	Mengamati metrik F1 dalam evaluasi model untuk melihat potensi ketidakseimbangan class pada data
Precision	Mengamati metrik Precision dalam evaluasi model untuk menunjukkan seberapa akurat prediksi positif model.
Recall	Mengamati metrik Recall dalam evaluasi model untuk menunjukkan seberapa baik model dalam menemukan semua contoh positif dalam dataset

4.4.3 Data Hasil Pengujian

4.4.3.1 *Blackbox Testing*

Tabel 4.4.3.1 Hasil *Blackbox Testing*

Modul	Skenario Uji	<i>Expected Result</i>	<i>Actual Result</i>	Kesimpulan
Login	Login dengan Kredensial Valid (Email dan Password)	Masuk ke halaman utama	Masuk ke halaman utama	Sesuai
	Login dengan	Pemberitahuan kesalahan	Pemberitahuan kesalahan	Sesuai

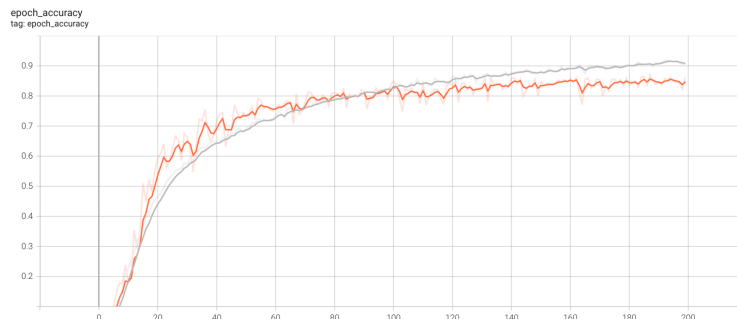
	Email Salah	kredensial	kredensial	
	Login dengan Password Salah	Pemberitahuan kesalahan kredensial	Pemberitahuan kesalahan kredensial	Sesuai
	Login dengan Kredensial Salah (Email dan Password)	Pemberitahuan kesalahan kredensial	Pemberitahuan kesalahan kredensial	Sesuai
<i>Word Level Gesture Recognition</i>	Memeragakan Gerakan Kata Bahasa Isyarat Sesuai dengan Kamus SIBI	Model mengeluarkan output label yang sesuai	Model mengeluarkan output label yang sesuai	Sesuai
	Memeragakan Gerakan Kata Bahasa Isyarat Tidak Sesuai dengan Kamus SIBI	Model mengeluarkan output label yang mendekati dengan input	Model mengeluarkan output label yang mendekati dengan input	Sesuai
	Memeragakan Gerakan Kata Bahasa Isyarat yang Tidak Ada di dalam Kamus SIBI	Model mengeluarkan output label yang mendekati dengan input	Model mengeluarkan output label yang mendekati dengan input	Sesuai

<i>Letter Level Gesture Recognition</i>	Memeragakan Gerakan Huruf Bahasa Isyarat Sesuai dengan Kamus SIBI	Model mengeluarkan output label yang mendekati dengan input	Model mengeluarkan output label yang mendekati dengan input	Sesuai
	Memeragakan Gerakan Huruf Bahasa Isyarat Tidak Sesuai dengan Kamus SIBI	Model mengeluarkan output label yang mendekati dengan input	Model mengeluarkan output label yang mendekati dengan input	Sesuai
	Memeragakan Gerakan Huruf Bahasa Isyarat yang Tidak Ada di dalam Kamus SIBI	Model mengeluarkan output label yang mendekati dengan input	Model mengeluarkan output label yang mendekati dengan input	Sesuai
<i>Hate Speech Detection</i>	Memeragakan atau Menyebutkan Ujaran Kebencian atau Bahasa Kasar	Pemberitahuan Bahasa Kasar atau Ujaran Kebencian	Pemberitahuan Bahasa Kasar atau Ujaran Kebencian	Sesuai
	Memeragakan atau Menyebutkan Kalimat atau Kata Biasa	Pemberitahuan Bahasa Kasar atau Ujaran Kebencian	Pemberitahuan Bahasa Kasar atau Ujaran Kebencian	Sesuai

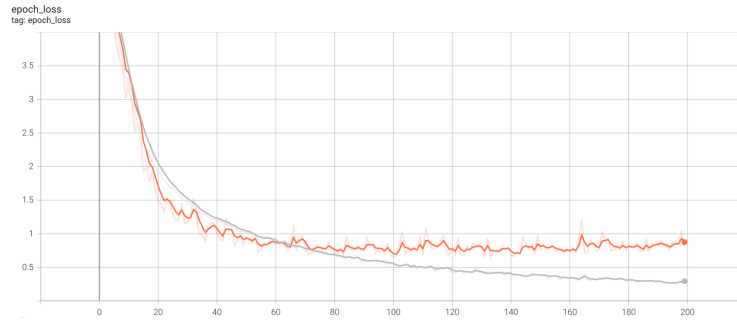
Kamus	Melihat Video Referensi Bahasa Isyarat dengan Spesifik	Mengembalikan Video Referensi dengan Labelnya	Mengembalikan Video Referensi dengan Labelnya	Sesuai
	Melakukan Advance Search Menggunakan Search Bar	Mengembalikan Video Referensi dengan Labelnya	Mengembalikan Video Referensi dengan Labelnya	Sesuai
	Melakukan Advance Search tidak sesuai dengan Kamus yang ada	Pemberitahuan Referensi Tidak Ditemukan	Pemberitahuan Referensi Tidak Ditemukan	Sesuai

4.4.3.2 Analisis Model Machine Learning

1. Analisis Model *Word Level Gesture Recognition*



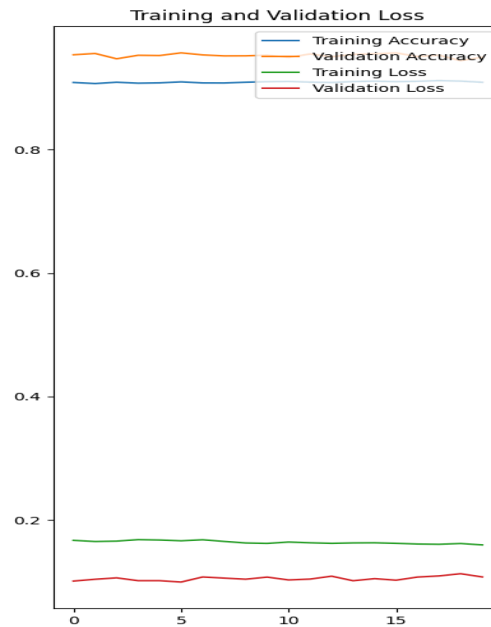
Gambar 4.4.3.2.2.1 Hasil Akurasi per Epoch



Gambar 4.4.3.2.2.2 Hasil Loss per Epoch

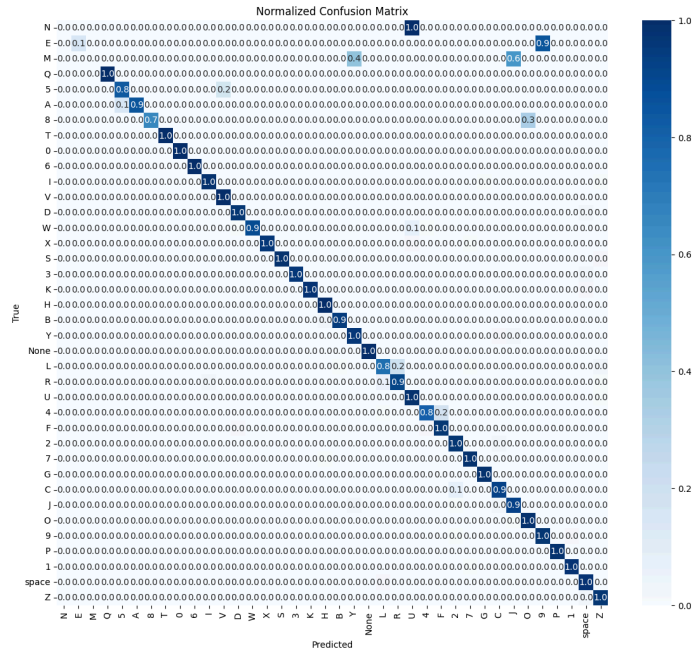
Model kami menghasilkan akurasi sebesar 90% dan akurasi validasi sebesar 86.3%. Hasil ini menunjukkan bahwa model memiliki kemampuan yang baik dalam mengenali gerakan tangan pada data latih dan data validasi. Akurasi yang tinggi pada data latih dan validasi menunjukkan bahwa model tidak mengalami overfitting secara signifikan, meskipun ada perbedaan akurasi yang menunjukkan sedikit overfitting. Selain itu, nilai loss pada data latih adalah 0.3387 dan pada data validasi adalah 0.7948. Perbedaan ini menunjukkan bahwa model mungkin perlu sedikit lebih dioptimalkan untuk mengurangi gap antara loss pada data latih dan validasi, namun secara keseluruhan performa model masih dalam kategori baik.

2. Analisis Model *Letter Level Gesture Recognition*



Gambar 4.4.3.2.1 Grafik Analisis per-Epoch

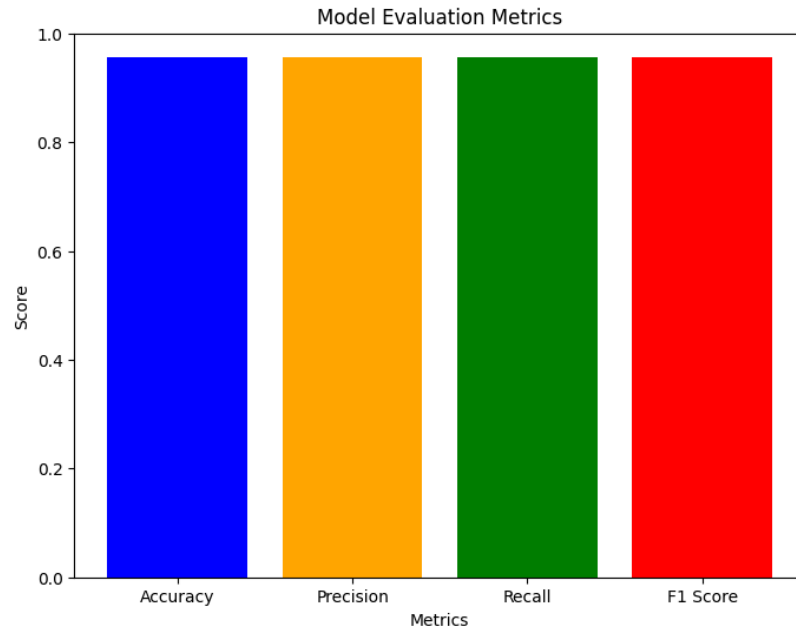
Dapat dilihat pada gambar, dengan fine-tuning yang tepat akan menghasilkan peningkatan akurasi yang cepat. Dengan menggunakan 20 *epoch gap* antara *loss* dan *accuracy* baik itu *training* maupun *validation* cukup jauh sehingga menghasilkan model yang tergolong bagus.



Gambar 4.4.3.2.2 Hasil Confussion Matrix

Confusion matrix adalah alat yang digunakan untuk mengevaluasi kinerja model klasifikasi. Ini memberikan informasi tentang prediksi yang benar dan salah yang dibuat oleh model untuk setiap kelas dalam dataset.

Confusion matrix yang digunakan seperti terlihat pada gambar adalah versi yang telah dinormalisasi, di mana setiap angka di dalam matriks mewakili proporsi dari instance yang diprediksi untuk setiap kelas dengan range 0 hingga 1. Ini memungkinkan evaluasi yang lebih mudah untuk dibaca. Secara keseluruhan, model menunjukkan kinerja yang sangat baik dengan sebagian besar kelas memiliki nilai pada diagonal utama yang tinggi. Namun, beberapa kelas menunjukkan adanya kesalahan klasifikasi yang cukup signifikan berdasarkan matriks tersebut, walaupun setelah dilakukannya pengujian secara *live stream* akurasi kelas tersebut cukup tinggi.

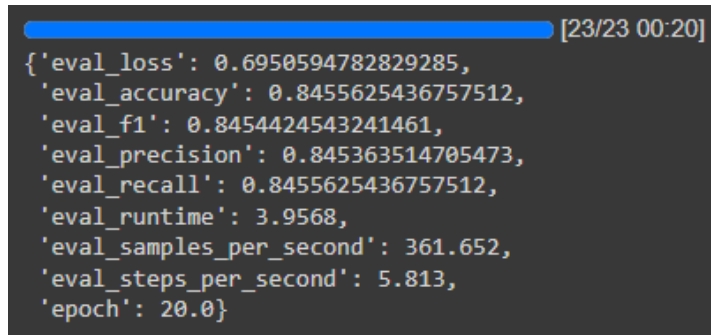


Gambar 4.4.3.2.3 Hasil Matrix Evaluasi

Gambar diatas menunjukkan 4 metrik evaluasi yang digunakan untuk menguji model, dimana memiliki nilai yang sama-sama tinggi, dengan Classification Report:

1. Accuracy: 0.9562
2. Precision: 0.9569
3. Recall: 0.9562
4. F1 Score: 0.9552

3. Analisis Model *Hate Speech Detection*



```
[23/23 00:20]
{'eval_loss': 0.6950594782829285,
 'eval_accuracy': 0.8455625436757512,
 'eval_f1': 0.8454424543241461,
 'eval_precision': 0.845363514705473,
 'eval_recall': 0.8455625436757512,
 'eval_runtime': 3.9568,
 'eval_samples_per_second': 361.652,
 'eval_steps_per_second': 5.813,
 'epoch': 20.0}
```

Gambar 4.4.3.3.1 Matrix Evaluasi

Dari hasil gambar 4.4.3.3.1, dapat disimpulkan bahwa akurasi model hate-speech recognition mencapai 0.86 dan bisa dikatakan cukup baik. Akan tetapi, fit dari model hate-speech ini masih belum dinyatakan good fit karena perbedaan training dan val loss yang cukup tinggi menyebabkan model ini underfit. Dari metrik evaluasi F1 yang didapatkan, terbukti bahwa model hate-speech ini memiliki keseimbangan class yang cukup baik karena metrik evaluasi F1 berhasil mencapai 0.86. Begitu juga dengan metrik evaluasi lainnya yaitu precision dan Recall yang berhasil mencapai angka 0.86. Hal ini terbukti bahwa model ini cukup baik dalam menunjukkan seberapa akurat prediksi positif model.

BAB V

PENUTUP

5.1 Simpulan

Dalam penelitian ini, telah berhasil dirancang dan dibangun sebuah aplikasi penerjemah bahasa isyarat berbasis MediaPipe untuk meningkatkan aksesibilitas komunikasi bagi tunarungu. Adapun simpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. **Keberhasilan Implementasi Teknologi MediaPipe:** Aplikasi yang dikembangkan mampu mendeteksi dan mengenali gestur bahasa isyarat dengan menggunakan teknologi MediaPipe secara real-time. Hal ini menunjukkan bahwa MediaPipe dapat diandalkan sebagai solusi dalam pengenalan gerakan tangan.
2. **Peningkatan Aksesibilitas Komunikasi:** Aplikasi ini terbukti dapat meningkatkan aksesibilitas komunikasi bagi tunarungu dengan menyediakan terjemahan bahasa isyarat ke teks secara cepat dan akurat. Ini membantu mengurangi hambatan komunikasi antara tunarungu dan orang yang tidak memahami bahasa isyarat.
3. **Desain Antarmuka Pengguna yang User-Friendly:** Aplikasi ini dirancang dengan antarmuka yang intuitif dan mudah digunakan, yang memudahkan pengguna tunarungu dalam mengoperasikan aplikasi tanpa memerlukan pelatihan khusus.
4. **Kinerja Sistem yang Efisien:** Pengujian terhadap aplikasi menunjukkan bahwa sistem ini memiliki kinerja yang efisien, mampu memproses video dan menghasilkan terjemahan dalam waktu yang singkat tanpa ada keterlambatan yang signifikan.

5.2 Saran

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut:

1. **Peningkatan Akurasi Pengenalan Gestur:** Diperlukan penelitian lebih lanjut untuk meningkatkan akurasi pengenalan gestur, terutama untuk gestur yang kompleks atau mirip satu sama lain. Penggunaan dataset yang lebih besar dan beragam dapat

membantu meningkatkan kemampuan sistem dalam mengenali berbagai gestur bahasa isyarat.

2. **Penambahan Fitur Baru:** Pengembangan aplikasi ini dapat dilanjutkan dengan menambahkan fitur-fitur baru, seperti terjemahan bahasa isyarat ke suara, dukungan untuk berbagai bahasa isyarat internasional, dan integrasi dengan perangkat wearable untuk meningkatkan interaktivitas.

3. **Optimalisasi Kinerja Aplikasi:** Perlu dilakukan optimasi lebih lanjut pada kinerja aplikasi untuk memastikan aplikasi tetap responsif dan efisien pada berbagai jenis perangkat, termasuk perangkat dengan spesifikasi rendah.

4. **Pengujian dengan Pengguna Sebenarnya:** Untuk memastikan aplikasi benar-benar memenuhi kebutuhan pengguna, disarankan untuk melakukan pengujian lapangan dengan melibatkan tunarungu sebagai pengguna sebenarnya. Umpan balik dari pengguna dapat digunakan untuk memperbaiki dan menyempurnakan aplikasi.

5. **Keamanan dan Privasi Data:** Penting untuk memperhatikan aspek keamanan dan privasi data pengguna, terutama data video yang direkam. Implementasi enkripsi data dan kebijakan privasi yang ketat perlu diterapkan untuk melindungi pengguna.

Dengan mengikuti saran-saran tersebut, diharapkan aplikasi penerjemah bahasa isyarat berbasis MediaPipe ini dapat terus berkembang dan memberikan kontribusi yang signifikan dalam meningkatkan aksesibilitas komunikasi bagi tunarungu.

DAFTAR PUSTAKA

- Alfikri, R. H. (2022, Juli). PENERJEMAH BAHASA ISYARAT DENGAN METODE CNN. *PEMBANGUNAN APLIKASI PENERJEMAH BAHASA ISYARAT DENGAN METODE CNN BERBASIS ANDROID*, Vol 16, No 2. 2615-224X
- Belajar Bahasa Isyarat*. (2023, December 6). Kementerian Pendidikan dan Kebudayaan » Republik Indonesia. Retrieved June 26, 2024, from <https://www.kemdikbud.go.id/main/blog/2023/12/kemendikbudristek-gelar-kelas-akhir-pekan-belajar-bahasa-isyarat>
- Cloud Firestore*. (n.d.). Firebase. Retrieved June 26, 2024, from <https://firebase.google.com/products/firestore>
- Gesture recognition task | Google for Developers*. (2024, May 21). Gemini API. Retrieved June 26, 2024, from https://ai.google.dev/edge/mediapipe/solutions/vision/gesture_recognizer
- Goyal, K., & G, V. (2023, April 20). INDIAN SIGH LANGUAGE RECOGNITION USING MEDIAPIPE HOLISTIC. *INDIAN SIGH LANGUAGE RECOGNITION USING MEDIAPIPE HOLISTIC*, 16. arXiv:2304.10256
- Holistic landmarks detection task guide | Google AI Edge | Google for Developers*. (2024, April 24). Gemini API. Retrieved June 27, 2024, from https://ai.google.dev/edge/mediapipe/solutions/vision/holistic_landmarker
- Patihullah, J., & Winarko, E. (2019, January 29). Hate Speech Detection forIndonesia Tweets UsingWord Embedding And Gated Recurrent Unit. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 13(1), 43-52. <https://doi.org/10.22146/ijccs.40125>

Riadi, M. (2020, July 25). *Tunarungu (Pengertian, Jenis, Penyebab, Karakteristik dan Proses Komunikasi)*. KajianPustaka. Retrieved June 26, 2024, from

<https://www.kajianpustaka.com/2020/07/tunarungu.html>

What is Machine Learning? (2024, May 14). Google for Developers. Retrieved June 26, 2024, from

<https://developers.google.com/machine-learning/intro-to-ml/what-is-ml>

What Is Python Used For? (2024, April 3). Coursera. Retrieved June 26, 2024, from

<https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>