

**INTEGRASI MODEL *DEEP LEARNING* DENGAN *PRE-TRAINED
MOBILENETV2* DAN *SPEECH RECOGNITION* PADA APLIKASI
VIDEO CALL ELCUE UNTUK PENDETEKSIAN BAHASA
ISYARAT INDONESIA (BISINDO)**

TUGAS AKHIR DALAM BENTUK LAIN SETARA SKRIPSI

HELGA PRICILLA BR. PURBA

211401067



**PROGRAM STUDI S1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2025**

**INTEGRASI MODEL *DEEP LEARNING* DENGAN *PRE-TRAINED
MOBILENETV2* DAN *SPEECH RECOGNITION* PADA APLIKASI
VIDEO CALL ELCUE UNTUK PENDETEKSIAN BAHASA
ISYARAT INDONESIA (BISINDO)**

TUGAS AKHIR DALAM BENTUK LAIN SETARA SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat untuk memperoleh ijazah
Sarjana Ilmu Komputer

HELGA PRICILLA BR. PURBA

211401067



**PROGRAM STUDI S1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2025**

PERSETUJUAN

Judul : INTEGRASI MODEL *DEEP LEARNING* DENGAN *PRE-TRAINED MOBILENETV2* DAN *SPEECH RECOGNITION* PADA APLIKASI *VIDEO CALL ELCUE* UNTUK PENDETEKSIAN BAHASA ISYARAT INDONESIA (BISINDO)

Kategori : TUGAS AKHIR DALAM BENTUK LAIN SETARA SKRIPSI

Nama : HELGA PRICILLA BR PURBA

Nomor Induk Mahasiswa : 211401067

Program Studi : S-1 ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Komisi Pembimbing :

Medan, 27 Maret 2025

Pembimbing II

Dr. Maya Silvi Lydia B.Sc., M.Sc.

NIP. 197401272002122001

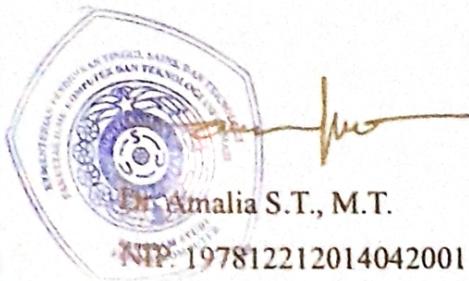
Pembimbing I

Hayatunnufus, S.Kom., M.Cs

NIP. 199207192024062001

Diketahui/disetujui oleh

Program Studi S-1 Ilmu Komputer



PERNYATAAN

INTEGRASI MODEL DEEP LEARNING DENGAN PRE-TRAINED
MOBILENETV2 DAN SPEECH RECOGNITION PADA APLIKASI
VIDEO CALL ELCUE UNTUK PENDETEKSIAN BAHASA
ISYARAT INDONESIA (BISINDO)

TUGAS AKHIR DALAM BENTUK LAIN SETARA SKRIPSI

Saya mengakui bahwa tugas akhir dalam bentuk lain setara skripsi ini merupakan hasil kerja sama tim dalam kompetisi. Penulisan judul dan isi dari tugas akhir dalam bentuk lain setara skripsi ini dilakukan secara individu dan berbeda satu dengan lainnya, yang disesuaikan dengan bidang masing-masing dalam tim, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 27 Maret 2025



Helga Pricilla Br. Purba

211401067

PENGHARGAAN

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat dan penyertaan-Nya, sehingga penulis dapat menyelesaikan bentuk lain setara skripsi yang berjudul “Integrasi Model *Deep Learning* dengan *Pre-trained MobileNetV2* dan *Speech Recognition* pada Aplikasi *Video Call ElCue* untuk Pendekripsi Bahasa Isyarat Indonesia (BISINDO)” sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada program studi S-1 Ilmu Komputer di Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

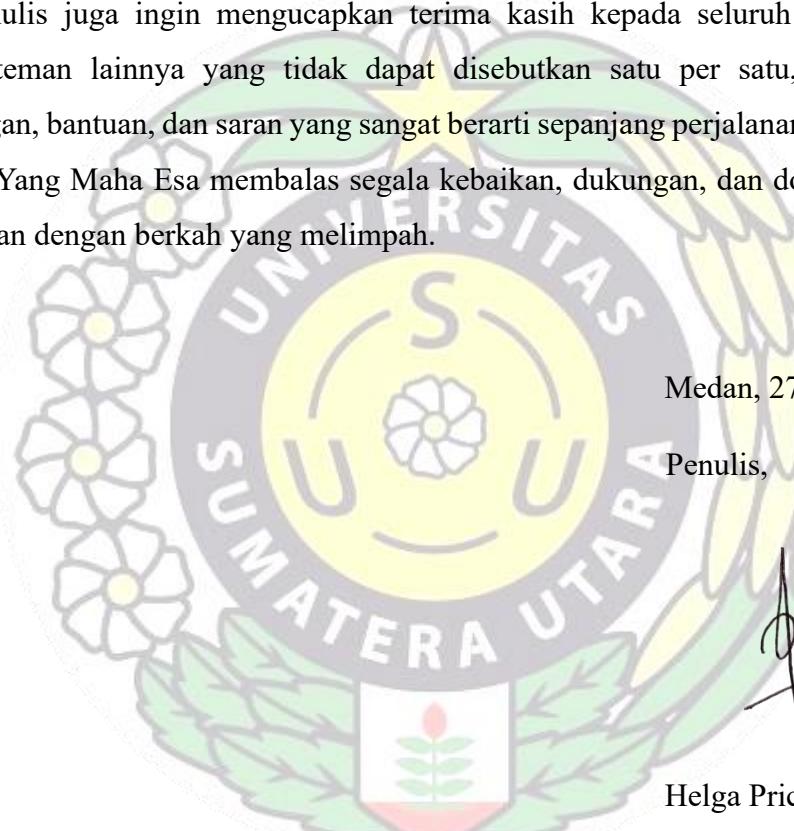
Penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan, bantuan, dan doa dalam menyelesaikan bentuk lain setara skripsi ini. Teristimewa, penulis ingin menyampaikan penghargaan kepada kedua orang tua tercinta, Ibu Lilys Siregar dan Bapak Ganda Purba, yang dengan sabar memberikan kasih sayang, waktu, dan perhatian tanpa henti dalam setiap langkah penulis. Terima kasih atas segala dukungan yang telah diberikan, yang sangat berarti bagi penulis dalam menyelesaikan tugas ini. Selain itu, penulis juga mengucapkan terima kasih yang mendalam kepada:

1. Tuhan Yesus Kristus, atas kasih dan berkat-Nya yang tiada henti. Dengan penyertaan-Nya, penulis dapat menyelesaikan bentuk lain setara skripsi ini dengan baik.
2. Bapak Prof. Dr. Muryanto Amin S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
3. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara, Dosen Pembimbing II dan Dosen Pembimbing Akademik yang telah memberikan banyak dukungan, motivasi, serta bimbingan yang sangat berarti, baik secara akademis maupun spiritual, selama proses penyusunan bentuk lain setara skripsi ini.
4. Bapak Dr. Mohammad Andri Budiman S.T., M.Comp.Sc., M.E.M. selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

5. Ibu Dr. Amalia, S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
6. Ibu Hayatunnufus S.Kom, M.Cs, selaku Dosen Pembimbing I dan Dosen Pendamping PKM, atas segala kesabaran, dukungan, dan motivasi yang Ibu berikan sepanjang proses penggeraan proyek ElCue dan penyusunan bentuk lain setara skripsi ini. Terima kasih atas bimbingan yang penuh arahan konstruktif dan semangat yang telah Ibu tanamkan, yang membuat penulis mampu melewati setiap langkah dengan lebih percaya diri.
7. Bapak dan Ibu Dosen Program Studi S1 Ilmu Komputer Universitas Sumatera Utara yang telah dengan tulus membagikan ilmu, wawasan, dan pengalaman berharga selama masa perkuliahan.
8. Staf dan pegawai Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang dengan sigap dan penuh perhatian telah membantu urusan administrasi perkuliahan hingga penyelesaian bentuk lain setara skripsi ini.
9. Abang penulis, Kevin Nabot, dan adik, Samuel Wilson, atas segala dukungan, kasih sayang, dan semangat yang tiada henti sepanjang perjalanan penulis, yang selalu menjadi penyemangat dan kekuatan, dan memberikan motivasi untuk terus maju.
10. Sahabat-sahabat Potobut, Tessa Agitha, Angela Siadari, Imanda Tamara, Putri Andriyani, dan Rani Widya, yang telah menemani penulis sepanjang perjalanan perkuliahan ini. Tidak hanya sebagai sahabat, tetapi juga sebagai keluarga yang selalu memberikan dukungan, semangat, dan kebahagiaan tanpa henti. Terima kasih atas kebersamaan dalam menjalani setiap suka dan duka, atas tawa yang menemani saat-saat penuh tantangan, dan atas motivasi yang tidak pernah surut. Kehadiran kalian sangat berarti dalam setiap langkah penulis, memberikan kekuatan dan inspirasi yang tak ternilai.
11. Sahabat penulis dari masa SMP, Sere Tania, Angel Aurora, dan Theofany Dwiyanty terima kasih atas dukungan, semangat, dan persahabatan yang telah diberikan, yang sangat membantu penulis sepanjang perjalanan ini.

12. Teman-teman dari Program Kreativitas Mahasiswa (PKM) hingga PIMNAS 37 kontingen Universitas Sumatera Utara, yang telah bersama penulis dalam setiap tahap kegiatan, serta atas dukungan dan kerja sama yang luar biasa sepanjang perjalanan kegiatan PKM dan PIMNAS.
13. Teman-teman Kom B dan stambuk 2021 terkhusus Tiur, Sindhy, Sri, Elisa, Mutia, Pieter, Haikal, Harry, Samuel Reysha dan Lorenzo yang telah memberikan peran selama masa perkuliahan di program studi Ilmu Komputer.

Penulis juga ingin mengucapkan terima kasih kepada seluruh kerabat dan teman-teman lainnya yang tidak dapat disebutkan satu per satu, atas segala dukungan, bantuan, dan saran yang sangat berarti sepanjang perjalanan ini. Semoga Tuhan Yang Maha Esa membalas segala kebaikan, dukungan, dan doa yang telah diberikan dengan berkah yang melimpah.



Medan, 27 Maret 2025

Penulis,



Helga Pricilla Br Purba

211401067

ABSTRAK

Penelitian ini berfokus pada integrasi model *deep learning* MobileNetV2 dan teknologi *Speech Recognition* dalam aplikasi *video call* untuk mendeteksi Bahasa Isyarat Indonesia (BISINDO) dan mengonversi suara menjadi teks secara *real-time*. Tujuan utama penelitian ini adalah untuk memfasilitasi komunikasi dua arah yang lebih inklusif antara penyandang disabilitas Tuli dan non-Tuli dalam konteks *video call*. Aplikasi ElCue mengintegrasikan kedua teknologi ini, dengan MobileNetV2 digunakan untuk mendeteksi gestur tangan pengguna Tuli yang diterjemahkan menjadi teks, sementara *Speech Recognition* digunakan untuk mentranskripsi ucapan pengguna non-Tuli menjadi teks. Model MobileNetV2, yang telah diubah menjadi format TensorFlow Lite (TFLite), berhasil mendeteksi gestur BISINDO dengan akurasi rata-rata 85,56%. Di sisi lain, teknologi *Speech Recognition*, melalui integrasi *Speech-to-Text API*, mentranskripsi suara dengan akurasi rata-rata 93%. Penggunaan Agora SDK untuk *video call* memastikan kualitas komunikasi audio dan video tetap lancar meskipun menggunakan teknologi AI yang memerlukan pemrosesan *real-time*. Hasil pengujian menunjukkan bahwa integrasi ini berhasil meningkatkan aksesibilitas komunikasi antara pengguna Tuli dan non-Tuli, meskipun masih ada tantangan terkait keterbatasan daya komputasi perangkat *mobile*. Penelitian ini memberikan kontribusi penting dalam pengembangan teknologi komunikasi yang lebih inklusif untuk penyandang disabilitas Tuli.

Kata Kunci: Integrasi MobileNetV2, *Speech Recognition*, *Video Call*, Bahasa Isyarat Indonesia (BISINDO), TensorFlow Lite, Agora SDK.

**INTEGRATION OF DEEP LEARNING MODEL WITH PRE-TRAINED
MOBILENETV2 AND SPEECH RECOGNITION IN ELCUE
VIDEO CALL APPLICATION FOR DETECTION OF
INDONESIAN SIGN LANGUAGE (BISINDO)**

ABSTRACT

This research focuses on the integration of the MobileNetV2 deep learning model and Speech Recognition technology in a video call application to detect Indonesian Sign Language (BISINDO) gestures and convert speech-to-text in real-time. The main objective of this study is to facilitate more inclusive two-way communication between Deaf and non-Deaf users in the context of video calls. The ElCue application integrates both technologies, with MobileNetV2 used to detect the hand gestures of Deaf users and translate them into text, while Speech Recognition is used to transcribe the speech of non-Deaf users into text. The MobileNetV2 model, converted into TensorFlow Lite (TFLite) format, successfully detected BISINDO gestures with an average accuracy of 85.56%. Meanwhile, Speech Recognition technology, through the integration of the Speech-to-Text API, transcribed speech with an average accuracy of 93%. The use of Agora SDK for video calls ensures smooth audio and video communication, despite the real-time processing requirements of AI technologies. Testing results show that this integration successfully enhances communication accessibility between Deaf and non-Deaf users, although challenges remain regarding the limited computational power of mobile devices. This study makes a significant contribution to the development of more inclusive communication technologies for the Deaf community.

Keywords: MobileNetV2 Integration, Speech Recognition, Video Call, Indonesian Sign Language (BISINDO), TensorFlow Lite, Agora SDK.

DAFTAR ISI

PERSETUJUAN.....	i
PERNYATAAN.....	ii
PENGHARGAAN.....	iii
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Penelitian	5
1.6.1 Studi Literatur	5
1.6.2 Analisis Permasalahan	5
1.6.3 Perancangan Sistem	5
1.6.4 Implementasi	6
1.6.5 Pengujian Sistem.....	6
1.6.6 Penyusunan Laporan	6
1.7 Sistematika Penulisan	6
BAB 2 LANDASAN TEORI	8
2.1 Deep Learning untuk Pendekripsi Citra	8

2.2	MobileNetV2.....	8
2.3	Speech Recognition	9
2.4	Integrasi MobileNetV2 dan <i>Speech Recognition</i> dalam Pendekripsi Bahasa Isyarat	10
2.5	Penelitian Terdahulu	11
BAB 3 ANALISIS DAN PERANCANGAN SISTEM		15
3.1	Analisis Sistem.....	15
3.1.1	Analisis Masalah	15
3.1.2	Analisis Data	17
3.1.3	Analisis Kebutuhan	18
3.1.4	Diagram Umum Sistem.....	21
3.1.5	Arsitektur Sistem.....	23
3.2	Pemodelan Sistem	24
3.2.1	Activity Diagram.....	24
3.2.2	Flowchart	26
BAB 4 IMPLEMENTASI DAN PENGUJIAN.....		29
4.1	Implementasi	29
4.1.1	Inisialisasi <i>Environment</i> Pengembangan.....	29
4.1.2	Implementasi MobileNetV2 untuk Pendekripsi BISINDO	33
4.1.3	Integrasi <i>Speech-to-Text API</i> untuk Pengenalan Suara.....	40
4.2	Pengujian Sistem	45
4.2.1	Hasil Pengujian Pendekripsi Gestur	45
4.2.2	Hasil Pengujian <i>Speech Recognition</i>	47
BAB 5 KESIMPULAN DAN SARAN		52
5.1	Kesimpulan	52
5.2	Saran	52

DAFTAR PUSTAKA	54
LAMPIRAN	56



DAFTAR TABEL

Tabel 3.1 Spesifikasi Perangkat Lunak.....	20
Tabel 3.2 Spesifikasi Perangkat Keras	20
Tabel 4.1 Spesifikasi Platform Pengujian Android.....	32
Tabel 4.2 Hasil Pengujian Pendekripsi Gestur	46



DAFTAR GAMBAR

Gambar 3.1 Diagram Ishikawa	17
Gambar 3.2 Diagram Umum Sistem	21
Gambar 3.3 Arsitektur Sistem	24
Gambar 3.4 Activity Diagram	26
Gambar 3.5 Flowchart	28
Gambar 4.1 Dependencies Implementasi TFLite	30
Gambar 4.2 Library Implementasi TFLite.....	30
Gambar 4.3 Dependencies Implementasi Agora SDK	31
Gambar 4.4 Library Implementasi Agora SDK.....	31
Gambar 4.5 Dependencies Implementasi Speech-to-Text API.....	32
Gambar 4.6 Library Implementasi Speech-to-Text API	32
Gambar 4.7 Code untuk Pemuatan dan Inferensi MobileNetV2 (TFLite)	35
Gambar 4.8 Code untuk Pengambilan Frame Video Setiap Pengguna Baru Bergabung	36
Gambar 4.9 Code untuk Pengambilan Frame Video Call Secara Berkala	37
Gambar 4.10 Code untuk Prapemrosesan Gambar.....	38
Gambar 4.11 Code untuk Inferensi dengan MobileNetV2	39
Gambar 4.12 Code untuk Menampilkan Teks Hasil Deteksi di UI Video Call ...	40
Gambar 4.13 Code untuk Menginisialisasi Speech-to-Text	41
Gambar 4.14 Code untuk Mendengarkan Suara dan Mengonversi ke Teks.....	42
Gambar 4.15 Code untuk Menampilkan Transkripsi di Layar	43
Gambar 4.16 Code untuk Menangani Transkripsi Timeout	44
Gambar 4.17 Code untuk Memulai dan Menghentikan Pengenalan Suara.....	45
Gambar 4.18 Tampilan Hasil Pengujian Speech Recognition.....	48
Gambar 4.19 Code untuk Pengujian Akurasi Speech Recognition	51

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang terus berkembang, komunikasi menjadi aspek penting dalam kehidupan sehari-hari. Teknologi *video call* telah memungkinkan orang untuk berkomunikasi secara *real-time* melalui audio dan visual, menghilangkan batasan jarak dan waktu. Namun, kemajuan teknologi tersebut belum dapat dirasakan oleh semua kalangan masyarakat, salah satunya adalah penyandang disabilitas Tuli. Menurut data Badan Pusat Statistik Nasional tahun 2022, jumlah penyandang Tuli di Indonesia mencapai 2,1 juta orang dari total penduduk sekitar 275,77 juta jiwa. Bagi penyandang Tuli, komunikasi melalui *video call* menghadapi tantangan signifikan karena ketergantungan pada suara dan kurangnya dukungan fitur pendekripsi Bahasa Isyarat Indonesia (BISINDO). Akibatnya, penyandang Tuli mengalami kesulitan dalam berkomunikasi secara efektif pada situasi krusial, seperti rapat, pembelajaran daring, atau konsultasi kesehatan, yang pada akhirnya memperlebar kesenjangan komunikasi antara pengguna Tuli dan pengguna non-Tuli. Untuk mengatasi hal ini, bahasa isyarat menjadi sarana komunikasi utama bagi penyandang Tuli dalam membentuk interaksi sosial.

Bahasa Isyarat Indonesia (BISINDO) adalah bahasa isyarat yang dipelajari secara alami oleh penyandang Tuli, yang dengan mudah dapat digunakan dalam pergaulan isyarat Tuli sehari-hari (Gumelar dkk., 2018). Kurangnya pengetahuan mengenai komunikasi menggunakan bahasa isyarat, khususnya BISINDO, pada kelompok non-Tuli menyebabkan tidak tercapainya komunikasi yang inklusif antara kelompok Tuli dan non-Tuli. Ketidakmampuan untuk berkomunikasi secara efektif dapat menyebabkan kesalahpahaman, kebingungan, dan bahkan isolasi sosial bagi individu yang Tuli.

Untuk mendukung integrasi teknologi dalam komunikasi yang lebih adil dan inklusif, ElCue hadir sebagai aplikasi *video call* yang bertujuan mengurangi kesenjangan teknologi serta meningkatkan aksesibilitas. Aplikasi ini

memungkinkan interaksi yang lebih efektif antara teman Tuli dan non-Tuli dengan fitur deteksi Bahasa Isyarat Indonesia (BISINDO) dalam *video call*. Dengan mengintegrasikan model *deep learning pre-trained* MobileNetV2 berbasis computer vision ke dalam aplikasi *mobile*, ElCue dapat mendeteksi dan menerjemahkan gerakan BISINDO secara *real-time* selama *video call* berlangsung. Sistem ini mengenali isyarat tangan pengguna, lalu mengonversinya menjadi teks yang ditampilkan di layar lawan bicara. Selain itu, ElCue juga dilengkapi fitur konversi suara ke teks, sehingga menciptakan komunikasi dua arah yang lebih inklusif antara pengguna Tuli dan non-Tuli.

Berbagai penelitian telah dilakukan untuk meningkatkan akurasi dan efisiensi model *deep learning* untuk diterapkan pada perangkat seluler. Salah satu penelitian yang relevan adalah studi yang dilakukan oleh Alfikri dkk (2022) dalam jurnal Pembangunan Aplikasi Penerjemah Bahasa Isyarat Dengan Metode CNN Berbasis Android. Penelitian ini menggunakan model *pre-trained* EfficientNet-Lite4, yang dioptimalkan untuk perangkat seluler karena menawarkan keseimbangan antara akurasi dan efisiensi komputasi. Meskipun model ini memiliki akurasi lebih tinggi dibandingkan varian lainnya, latensi yang dihasilkan masih menjadi tantangan dalam implementasi *real-time*. Hasil penelitian menunjukkan bahwa setelah model diintegrasikan ke dalam aplikasi *mobile*, akurasinya menurun dari 95% menjadi 73%, dengan beberapa isyarat mengalami kesalahan prediksi. Hal ini menunjukkan bahwa meskipun model *deep learning* dapat diterapkan pada perangkat seluler, faktor seperti optimasi komputasi dan keterbatasan daya pemrosesan masih mempengaruhi performanya.

Dalam konteks ini, penelitian yang dilakukan pada aplikasi ElCue berfokus pada integrasi *pre-trained* MobileNetV2 dan teknologi *speech recognition* untuk mendeteksi Bahasa Isyarat Indonesia (BISINDO) dalam lingkungan *video call* secara *real-time*. MobileNetV2 dipilih karena memiliki efisiensi komputasi yang tinggi dengan jumlah parameter lebih ringan dibandingkan EfficientNet-Lite4, sehingga lebih cocok untuk aplikasi yang berjalan pada perangkat dengan keterbatasan daya komputasi. Selain itu, integrasi dengan *speech recognition* memungkinkan komunikasi dua arah yang lebih inklusif, di mana suara dapat

dikonversi menjadi teks dan bahasa isyarat dapat dideteksi serta diterjemahkan secara *real-time*.

Dengan mempertimbangkan tantangan akurasi yang telah diidentifikasi dalam penelitian sebelumnya, pengembangan ElCue bertujuan untuk menghadirkan solusi komunikasi yang lebih efisien bagi teman Tuli. Pendekatan yang diusulkan dalam penelitian ini menggabungkan keunggulan MobileNetV2 dalam deteksi gerakan dengan teknologi *speech recognition*, sehingga dapat meningkatkan aksesibilitas komunikasi dalam *video call* tanpa mengorbankan kinerja sistem. Dengan demikian, penelitian ini diharapkan dapat berkontribusi dalam mengatasi hambatan komunikasi antara individu Tuli dan non-Tuli melalui pendekatan berbasis *deep learning* yang lebih optimal untuk perangkat seluler.

1.2 Rumusan Masalah

Penelitian ini berfokus pada tantangan dalam mengintegrasikan model *deep learning* untuk deteksi Bahasa Isyarat Indonesia (BISINDO) dan pengenalan suara (*speech recognition*) untuk transkripsi suara ke dalam aplikasi *video call* secara *real-time*. Meskipun model *pre-trained* seperti MobileNetV2 menawarkan efisiensi komputasi yang baik, tantangan utama terletak pada pengintegrasian model MobileNetV2 ke dalam aplikasi *video call* untuk mendeteksi gerakan BISINDO secara *real-time*, terutama pada perangkat seluler yang memiliki keterbatasan sumber daya. Di sisi lain, teknologi *speech recognition* memiliki peran penting dalam mendukung komunikasi dengan menyediakan transkripsi otomatis dari suara menjadi teks yang dapat dimengerti. Hal ini memungkinkan komunikasi dua arah yang lebih efektif antara pengguna yang berbicara dan mereka yang menggunakan BISINDO. Oleh karena itu, penting untuk menganalisis bagaimana integrasi kedua teknologi ini dapat meningkatkan efektivitas serta kenyamanan komunikasi dalam aplikasi ElCue, dengan memaksimalkan potensi keduanya dalam lingkungan perangkat seluler yang terbatas..

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk menerapkan penggunaan model *deep learning* dengan *pre-trained* MobileNetV2 yang terintegrasi dengan teknologi

speech recognition pada aplikasi *video call* ElCue untuk pendekripsi Bahasa Isyarat Indonesia (BISINDO) secara *real-time* pada perangkat *mobile*.

1.4 Batasan Penelitian

Untuk mencegah meluasnya penelitian dan agar penelitian dapat lebih terarah, dibutuhkan adanya batasan masalah sebagaimana tertera pada poin-poin berikut:

1. Penelitian ini hanya mencakup integrasi model *deep learning* MobileNetV2 untuk pendekripsi gestur tangan yang merupakan isyarat dari Bahasa Isyarat Indonesia (BISINDO), serta teknologi *Speech Recognition* untuk mengonversi suara pengguna menjadi teks secara *real-time* dalam aplikasi *video call* ElCue. Kedua teknologi ini bekerja secara bersamaan untuk mendukung penerjemahan BISINDO dan memungkinkan interaksi *video call* yang lebih inklusif dan aksesibel.
2. Pengujian dilakukan untuk mengevaluasi akurasi deteksi BISINDO setelah model *deep learning* diintegrasikan dalam aplikasi *video call* serta akurasi *Speech-to-Text* dalam mengonversi suara pengguna ke teks dalam aplikasi. Pengujian ini mengukur efektivitas kedua sistem dalam kondisi penggunaan nyata pada perangkat *mobile*.
3. Dataset BISINDO yang digunakan dalam pelatihan model MobileNetV2 hanya terdiri dari 9 kelas yaitu “A”, “Asal”, “B”, “C”, “D”, “E”, “Halo”, “I”, “Terima Kasih”.
4. Implementasi aplikasi *mobile* menggunakan *framework* Flutter untuk pengembangan antarmuka dan TensorFlow Lite untuk integrasi model *deep learning* pada perangkat seluler.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini, yaitu:

1. Membantu mengetahui bagaimana pengoptimalan integrasi model *deep learning* MobileNetV2 dan teknologi *speech recognition* ke dalam aplikasi *video call* ElCue sehingga dapat mendekripsi dan menerjemahkan Bahasa Isyarat Indonesia (BISINDO) secara *real-time* dengan akurasi yang optimal.

2. Diharapkan pengguna, khususnya penyandang disabilitas Tuli dan pengguna lainnya, dapat berkomunikasi lebih efektif melalui aplikasi ElCue yang telah terintegrasi dengan model *deep learning* dan *speech recognition*, sehingga komunikasi dua arah antara pengguna Tuli dan non-Tuli dapat berjalan lebih lancar.
3. Menyediakan solusi teknologi berbasis *mobile* yang dapat membantu meningkatkan inklusivitas komunikasi bagi pengguna Tuli di berbagai situasi, seperti pendidikan, dunia kerja, dan interaksi sosial sehari-hari.

1.6 Metodologi Penelitian

Adapun tahapan yang dilalui untuk melakukan penelitian ini adalah sebagai berikut:

1.6.1 Studi Literatur

Mengumpulkan berbagai sumber referensi yang relevan dengan topik penelitian, seperti buku, jurnal, makalah, dan artikel yang membahas mengenai penerapan model *deep learning* MobileNetV2, *speech recognition*, dan teknologi yang digunakan dalam aplikasi *video call* untuk pendekripsi Bahasa Isyarat Indonesia (BISINDO). Sumber referensi ini akan memberikan dasar teori dan konsep yang diperlukan dalam penelitian.

1.6.2 Analisis Permasalahan

Tahap ini melibatkan analisis mendalam terhadap masalah yang dihadapi dalam pengintegrasian model *deep learning* MobileNetV2 dan *speech recognition* ke dalam aplikasi *video call*. Analisis ini mencakup pemahaman mengenai tantangan akurasi deteksi gerakan BISINDO, serta pengaruh integrasi *speech recognition* terhadap proses komunikasi.

1.6.3 Perancangan Sistem

Berdasarkan analisis masalah yang telah dilakukan, perancangan sistem dilakukan dengan merancang arsitektur aplikasi *video call* ElCue yang mengintegrasikan model *deep learning* MobileNetV2 untuk deteksi BISINDO dan teknologi *speech recognition* untuk konversi suara menjadi

teks. Desain sistem ini juga mencakup antarmuka pengguna (UI) yang mendukung interaksi yang inklusif.

1.6.4 Implementasi

Setelah perancangan sistem selesai, tahap implementasi dilakukan untuk mengintegrasikan model MobileNetV2 yang telah dilatih sebelumnya dengan aplikasi ElCue, serta mengimplementasikan teknologi *speech recognition* untuk konversi suara. Seluruh elemen sistem yang telah dirancang akan diimplementasikan pada platform *mobile* yang sesuai, menggunakan *framework* yang tepat untuk aplikasi *video call*.

1.6.5 Pengujian Sistem

Tahap pengujian dilakukan untuk mengukur performa dan akurasi sistem yang telah diimplementasikan. Pengujian mencakup evaluasi terhadap akurasi deteksi BISINDO pada aplikasi, serta kinerja konversi suara menjadi teks oleh *speech recognition*. Pengujian ini bertujuan untuk memastikan sistem bekerja dengan baik dan memberikan hasil yang sesuai dengan tujuan penelitian.

1.6.6 Penyusunan Laporan

Pada tahap akhir, dilakukan penyusunan laporan penelitian yang memaparkan seluruh proses yang telah dilakukan, termasuk studi literatur, analisis permasalahan, perancangan dan implementasi sistem, hasil pengujian, serta kesimpulan yang dapat diambil dari penelitian ini. Laporan ini juga akan memberikan rekomendasi untuk pengembangan lebih lanjut dalam aplikasi ElCue untuk meningkatkan inklusivitas komunikasi bagi pengguna BISINDO.

1.7 Sistematika Penulisan

Sistematika penulisan pada penelitian ini terdiri atas lima bab, sebagai berikut:

Bab 1: Pendahuluan

Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian, serta sistematika penulisan dari penelitian ini. Penelitian ini bertujuan untuk mengintegrasikan model *deep learning* dengan *pre-trained* MobileNetV2 dan *speech recognition* pada aplikasi *video call* ElCue untuk pendekripsi Bahasa Isyarat Indonesia (BISINDO).

Bab 2: Landasan Teori

Bab ini berisi teori-teori yang berkaitan dengan penelitian, termasuk teori-teori mengenai *deep learning*, model MobileNetV2, *speech recognition*, computer vision, dan deteksi Bahasa Isyarat Indonesia. Selain itu, teori terkait dengan aplikasi *video call*, pemrosesan sinyal suara, serta penerapan teknologi ini pada komunikasi inklusif juga dibahas dalam bab ini.

Bab 3: Analisis dan Perancangan Sistem

Bab ini berisi analisis kebutuhan sistem dan perancangan arsitektur sistem secara umum. Penjelasan mengenai tahapan-tahapan dalam perancangan sistem, termasuk pengintegrasian model *deep learning* dengan MobileNetV2 dan *speech recognition*, serta desain alur kerja aplikasi *video call* ElCue untuk pendekripsi BISINDO, disampaikan dalam bab ini.

Bab 4: Implementasi dan Pengujian Sistem

Bab ini menguraikan tentang implementasi dari perancangan yang telah dibahas pada bab sebelumnya, serta penjelasan mengenai proses integrasi model *deep learning* dengan *pre-trained* MobileNetV2 dan *speech recognition* pada aplikasi. Selain itu, bab ini juga memaparkan hasil pengujian sistem yang telah dibangun untuk menilai kinerja deteksi BISINDO dan akurasi sistem dalam mengenali isyarat serta suara.

Bab 5: Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan, serta saran-saran untuk penelitian lanjutan atau pengembangan lebih lanjut dari aplikasi ElCue dalam mendukung komunikasi inklusif menggunakan teknologi deteksi Bahasa Isyarat Indonesia (BISINDO).

BAB 2

LANDASAN TEORI

2.1 Deep Learning untuk Pendekripsi Citra

Deep learning adalah teknik dalam pembelajaran mesin yang menggunakan jaringan saraf tiruan dengan banyak lapisan untuk mengekstraksi pola dari data kompleks secara otomatis. Teknologi ini telah mengalami perkembangan pesat dalam bidang computer vision, memungkinkan sistem untuk mengenali, mendekripsi, dan mengklasifikasikan objek dalam gambar dan video tanpa pemrograman eksplisit (Purwono dkk., 2022). Salah satu arsitektur *deep learning* yang paling banyak digunakan dalam pendekripsi citra adalah *Convolutional Neural Networks* (CNN), yang dirancang untuk menangani data visual secara efisien. CNN terdiri dari beberapa lapisan utama, yaitu lapisan konvolusi untuk mengekstraksi fitur visual, lapisan *pooling* untuk mengurangi dimensi data dan meningkatkan efisiensi komputasi, serta lapisan *fully connected* untuk menghasilkan prediksi akhir. Fungsi aktivasi seperti ReLU (*Rectified Linear Unit*) juga digunakan untuk meningkatkan kinerja dan mempercepat konvergensi model (Taye, 2023).

CNN telah digunakan dalam berbagai aplikasi computer vision, termasuk pengenalan wajah, analisis gerakan, dan deteksi objek otomatis, menjadikannya solusi utama dalam pengembangan teknologi aksesibilitas bagi penyandang disabilitas pendengaran. Dengan keunggulannya dalam mengenali pola visual yang kompleks, *deep learning* menjadi dasar bagi pengembangan sistem komunikasi inklusif berbasis *video call* yang mengandalkan integrasi deteksi citra dan pengenalan suara untuk memperkaya pengalaman pengguna.

2.2 MobileNetV2

MobileNetV2 adalah arsitektur jaringan saraf konvolisional yang dirancang untuk meningkatkan efisiensi pemrosesan gambar pada perangkat dengan keterbatasan sumber daya, seperti ponsel dan perangkat *edge*. Model ini memperkenalkan struktur *inverted residual* dan *linear bottleneck* (Zhou dkk., 2020), yang

mengurangi jumlah parameter dan kompleksitas komputasi tanpa mengorbankan akurasi, menjadikannya ideal untuk tugas-tugas seperti klasifikasi gambar dan deteksi objek.

Untuk mengintegrasikan MobileNetV2 ke dalam aplikasi *mobile*, model yang telah dilatih dapat dikonversi menggunakan TensorFlow Lite (TFLite), sebuah *framework* yang memungkinkan implementasi model *machine learning* pada perangkat dengan sumber daya terbatas. Proses konversi ini melibatkan penggunaan TensorFlow Lite *Converter* untuk mengubah model yang telah dilatih ke dalam format .tflite, yang kemudian dapat diimplementasikan dalam aplikasi *mobile* (Alhanafi & Kurniawardhani, 2024).

Studi kasus implementasi MobileNetV2 dengan TFLite telah dilakukan dalam berbagai aplikasi. Misalnya, penelitian oleh Lu dkk. (2022) mengembangkan model MobileNetV2 yang dioptimalkan untuk identifikasi penyakit tanaman pada perangkat *edge* dan *mobile*. Hasil penelitian menunjukkan bahwa model yang dioptimalkan ini mencapai akurasi tinggi dengan jumlah parameter yang lebih sedikit, serta peningkatan kecepatan inferensi sebesar 8,5% dibandingkan dengan model asli, menjadikannya cocok untuk aplikasi *real-time* pada perangkat *mobile*.

2.3 *Speech Recognition*

Teknologi pengenalan suara (*Speech Recognition*) memainkan peran penting dalam interaksi antara manusia dan komputer dengan memungkinkan sistem mendeteksi dan mengonversi ucapan menjadi teks secara otomatis. Penerapan teknologi ini telah meluas ke berbagai sektor, termasuk asisten virtual, navigasi berbasis suara, dan alat komunikasi bagi individu dengan gangguan pendengaran (Ryumin dkk., 2023). Dalam ekosistem pengembangan aplikasi, Flutter menyediakan *Speech-to-Text API* melalui paket `speech_to_text`, yang memungkinkan aplikasi untuk melakukan pengenalan suara dalam mode *offline* maupun *online* menggunakan API pengenalan suara bawaan perangkat. Mode *online* sering kali mengandalkan layanan *cloud* seperti Google *Speech-to-Text API* untuk meningkatkan akurasi transkripsi. Teknologi ini telah diterapkan dalam berbagai skenario, termasuk

aplikasi *video call*, *chatbot* interaktif, serta sistem komunikasi yang mendukung aksesibilitas bagi pengguna dengan gangguan pendengaran.

Studi yang dilakukan oleh Nirbhavane dkk. (2020) menunjukkan bahwa penerapan *Speech-to-Text API* dalam Flutter dapat digunakan dalam aplikasi *video call* untuk menghasilkan transkripsi percakapan secara langsung, sehingga membantu pengguna yang memiliki gangguan pendengaran atau berada dalam lingkungan bising. Selain itu, penelitian Colón & Jonathan (2023) mengembangkan aplikasi berbasis Flutter untuk keperluan medis dan komunikasi darurat, yang memanfaatkan teknologi *Speech-to-Text* untuk mengonversi percakapan pasien menjadi teks, sehingga memudahkan dokumentasi serta rujukan medis. Dari segi implementasi, paket `speech_to_text` dalam Flutter terdiri dari beberapa komponen utama, yakni perekaman suara, pengolahan suara menjadi teks, serta tampilan teks secara *real-time* dalam antarmuka aplikasi. Dengan semakin berkembangnya teknologi AI dan *Natural Language Processing* (NLP), penggunaan *Speech-to-Text API* dalam Flutter diprediksi akan semakin canggih, memungkinkan fitur transkripsi yang lebih akurat, penerjemahan otomatis, serta integrasi dengan berbagai sistem berbasis suara untuk meningkatkan pengalaman pengguna.

2.4 Integrasi MobileNetV2 dan *Speech Recognition* dalam Pendekripsi Bahasa Isyarat

Integrasi MobileNetV2 dan *Speech Recognition* dalam pendekripsi bahasa isyarat bertujuan untuk menciptakan sistem komunikasi yang lebih inklusif bagi penyandang Tuli. MobileNetV2 adalah arsitektur *Convolutional Neural Network* (CNN) yang dirancang untuk mengenali gestur bahasa isyarat secara *real-time*, terutama pada perangkat dengan keterbatasan daya komputasi. Model ini menggunakan pendekatan *transfer learning*, yang memungkinkan pemanfaatan fitur dari dataset skala besar seperti *ImageNet*, sehingga meningkatkan akurasi dalam mengklasifikasikan berbagai gestur (Jagtap dkk., 2024). Sementara itu, *Speech Recognition* berperan dalam mengenali ucapan dan mengonversinya menjadi teks, sehingga memungkinkan interaksi yang lebih efektif dan mudah.

diakses oleh berbagai pengguna (Srivastava, 2024). Teknologi ini juga meningkatkan inklusivitas komunikasi dengan menjembatani interaksi antara individu dengan gangguan pendengaran dan mereka yang tidak mengalaminya.

Proses integrasi kedua teknologi ini diawali dengan *pre-processing* data, di mana citra tangan diekstraksi dari video menggunakan teknik segmentasi, sementara sinyal audio difilter untuk mengurangi *noise*, sehingga meningkatkan keakuratan pengenalan suara. MobileNetV2 kemudian memproses citra untuk mengidentifikasi dan mengklasifikasikan gestur berdasarkan fitur visual seperti orientasi tangan, bentuk jari, dan pola pergerakan. Secara bersamaan, *Speech Recognition* mengenali pola suara yang diucapkan bersama dengan gestur tersebut. Untuk memastikan model MobileNetV2 dapat berjalan secara efisien di perangkat *mobile*, model yang telah dilatih dikonversi ke format TensorFlow Lite (TFLite). TFLite memungkinkan model *deep learning* untuk berjalan dengan ukuran yang lebih kecil dan konsumsi daya yang lebih rendah, sehingga meningkatkan efisiensi inferensi dalam aplikasi berbasis *mobile*. Konversi ini dilakukan menggunakan TensorFlow Lite *Converter*, yang mengoptimalkan model melalui teknik seperti *quantization* untuk mengurangi ukuran tanpa mengorbankan akurasi (Alhanafi & Kurniawardhani, 2024). Dengan pendekatan ini, MobileNetV2 yang telah dikonversi ke TFLite dapat berjalan lebih cepat dalam aplikasi *video call*, memungkinkan pendekripsi gestur yang lebih responsif dan *real-time*. Kombinasi kedua teknologi ini memungkinkan penerjemahan yang lebih akurat dan kontekstual, sehingga makna bahasa isyarat dapat disampaikan dengan lebih jelas sesuai dengan konteks percakapan.

2.5 Penelitian Terdahulu

Penelitian yang dilakukan oleh Santoso dkk. (2024) membahas penerapan model *deep learning* untuk deteksi penyakit daun apel dengan menggunakan model ReXNet-150, EfficientNet, dan CNN konvensional. Penelitian ini mengintegrasikan model-model tersebut ke dalam aplikasi Android menggunakan TensorFlow Lite untuk mendekripsi penyakit daun apel secara *real-time* pada perangkat *mobile*. Untuk mengoptimalkan model agar dapat dijalankan pada

perangkat dengan keterbatasan komputasi, model yang sudah terlatih diubah ke dalam format TensorFlow Lite. Proses ini melibatkan konversi model yang dilatih menggunakan dataset gambar daun apel ke dalam format yang kompatibel dengan perangkat *mobile*, dan melakukan optimasi lebih lanjut menggunakan teknik seperti *quantization* dan *pruning* untuk mengurangi ukuran model dan mempercepat waktu inferensi. Aplikasi Android yang dikembangkan memungkinkan pengguna untuk mengambil foto atau mengunggah gambar daun apel, kemudian aplikasi ini mengklasifikasikan kondisi daun dalam waktu nyata, memberikan hasil diagnosis yang segera diterima pengguna. Penelitian ini memberikan wawasan tentang bagaimana mengatasi tantangan dalam mengintegrasikan model *deep learning* ke dalam aplikasi *mobile*, khususnya dalam hal optimasi ukuran model dan efisiensi inferensi untuk aplikasi dengan sumber daya terbatas.

Penelitian yang dilakukan oleh Jonathan & Rakun (2022) berfokus pada pengembangan aplikasi penerjemah bahasa isyarat Indonesia (SIBI) menjadi teks secara *real-time* menggunakan perangkat *mobile*. Sistem ini memanfaatkan beberapa teknologi pembelajaran mesin, di antaranya MobileNetV2 untuk ekstraksi fitur, *Conditional Random Field* (CRF) untuk mendeteksi gerakan transisi (epenthesis), dan *Long Short-Term Memory* (LSTM) untuk klasifikasi kata. Penelitian ini mengimplementasikan sistem terjemahan SIBI pada perangkat *mobile* dengan menggunakan metode inferensi perangkat di dalamnya untuk mengurangi penggunaan data. Hasil dari sistem ini menunjukkan tingkat akurasi kata mencapai 90,56% dan akurasi kalimat 64%, dengan waktu terjemahan rata-rata sekitar 31 detik. Penelitian ini juga mengimplementasikan teknik multi-inferensi paralel untuk mempercepat waktu terjemahan, yang berhasil mengurangi waktu terjemahan rata-rata hingga 25%. Dengan menggunakan pendekatan ini, penelitian ini berhasil mengatasi beberapa tantangan dalam pengenalan gerakan bahasa isyarat dan meningkatkan keefektifan aplikasi penerjemah SIBI berbasis *smartphone*.

Penelitian yang dilakukan oleh Rajput dkk. (2024) membahas penerapan model *deep learning* untuk klasifikasi penyakit daun tomat dengan menggunakan MobileNetV2 sebagai salah satu arsitektur dasar. Mereka mengintegrasikan model MobileNetV2, yang dikenal karena efisiensinya dalam komputasi dan ukuran

model yang ringan, ke dalam sistem berbasis *transfer learning* untuk memanfaatkan kemampuan model yang telah dilatih sebelumnya pada dataset besar seperti *ImageNet*. Dalam pengembangan aplikasi untuk pengenalan penyakit daun tomat, model MobileNetV2 digunakan karena kemampuannya untuk mengenali fitur dengan efisiensi tinggi, yang sangat sesuai untuk aplikasi berbasis perangkat *mobile* dengan keterbatasan sumber daya. Proses *fine-tuning* diterapkan pada lapisan akhir dari MobileNetV2 untuk disesuaikan dengan klasifikasi penyakit daun tomat, mengganti lapisan *fully connected* untuk mencocokkan kategori penyakit yang relevan. Hasil pengujian menunjukkan bahwa integrasi MobileNetV2 dalam aplikasi *mobile* untuk klasifikasi penyakit daun tomat memungkinkan deteksi penyakit dengan akurasi yang tinggi, bahkan dengan pengolahan pada perangkat *mobile* yang memiliki keterbatasan komputasi. Model ini berhasil mencapai akurasi lebih dari 96%, menunjukkan bahwa integrasi model MobileNetV2 memberikan kinerja yang efisien dan efektif dalam aplikasi pengenalan gambar berbasis *mobile*.

Penelitian yang dilakukan oleh Rao dkk. (2024) membahas integrasi teknologi pengenalan suara dan *machine learning* ke dalam *Online Public Access Catalogs* (OPACs) untuk layanan perpustakaan. Studi ini mengeksplorasi bagaimana teknologi pengenalan suara dapat meningkatkan aksesibilitas dan pengalaman pengguna dalam sistem perpustakaan dengan memungkinkan pengguna mencari bahan pustaka secara *hands-free*. Dengan mengintegrasikan pengenalan suara dengan *Natural Language Processing* (NLP) dan machine learning, sistem ini memungkinkan interaksi yang lebih intuitif dan berbasis suara, yang membantu pengguna menavigasi dan mengambil informasi dengan lebih efisien. Penelitian ini menyoroti penggunaan API seperti Google Cloud *Speech-to-Text* dan Microsoft Azure *Speech*, yang memungkinkan perpustakaan untuk dengan mudah menambahkan fitur suara ke sistem OPAC yang sudah ada. Integrasi ini bertujuan untuk mengatasi keterbatasan sistem tradisional, seperti antarmuka yang ketinggalan zaman dan kurangnya kompatibilitas dengan perangkat *mobile*, serta meningkatkan pengalaman pengguna melalui dukungan multibahasa dan hasil pencarian yang dipersonalisasi. Selain itu, penelitian ini juga menekankan pentingnya merancang perintah suara yang sederhana dan alami, sehingga

pengguna dari berbagai demografi dapat memanfaatkan teknologi ini secara efektif. Penelitian ini menegaskan bahwa teknologi pengenalan suara, yang didukung oleh AI dan NLP, dapat merevolusi cara pengguna berinteraksi dengan sistem katalog perpustakaan, menjadikannya lebih aksesibel, efisien, dan sesuai dengan harapan pengguna modern.

Penelitian yang dilakukan oleh Luo dkk. (2020) membahas perbandingan dan *benchmarking* berbagai model AI dan kerangka kerja pada perangkat *mobile*. Studi ini mencakup evaluasi model MobileNetV2 yang telah dikonversi ke format TensorFlow Lite (TFLite) untuk mengukur kinerja inferensi pada perangkat dengan sumber daya terbatas. Hasil penelitian menunjukkan bahwa penggunaan TFLite memungkinkan model MobileNetV2 berjalan lebih efisien pada perangkat *mobile*, dengan pengurangan ukuran model dan peningkatan kecepatan inferensi tanpa mengorbankan akurasi secara signifikan. Temuan ini menegaskan bahwa konversi model *deep learning* ke format TFLite merupakan langkah krusial dalam pengembangan aplikasi *mobile* yang memerlukan pemrosesan AI *real-time* dengan keterbatasan komputasi.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem ini membahas bagaimana model *deep learning* yang telah dioptimasi dan teknologi *speech recognition* dapat terintegrasi dalam aplikasi *video call* untuk memfasilitasi komunikasi dua arah antara pengguna Tuli dan pengguna non-Tuli. Fokus utama analisis adalah memahami bagaimana sistem menerima, memproses, dan menyajikan data secara *real-time* agar interaksi berlangsung secara alami dan efisien. Proses ini mencakup penerimaan input dari video dan audio, pemrosesan data untuk mengenali gerakan isyarat serta ucapan, hingga penyampaian hasil dalam bentuk teks atau suara yang dapat dipahami oleh lawan bicara.

Untuk memberikan pemahaman yang lebih jelas, gambaran umum sistem akan disajikan dalam bentuk diagram yang mengilustrasikan aliran data dan interaksi antar komponen utama dalam aplikasi. Diagram ini mencakup proses penerimaan data dari kamera dan mikrofon, pemrosesan menggunakan model *deep learning* yang telah diintegrasikan, serta konversi ucapan menjadi teks melalui *speech recognition*. Hasil pemrosesan ini kemudian disesuaikan dalam format yang mendukung komunikasi dalam *video call* agar tetap lancar dan efektif. Analisis ini menjadi dasar dalam pengembangan sistem yang tidak hanya efisien, tetapi juga mampu memastikan bahwa setiap komponen bekerja secara optimal untuk meningkatkan akurasi deteksi dan memperlancar interaksi antara pengguna dengan kebutuhan komunikasi yang berbeda.

3.1.1 Analisis Masalah

Bahasa Isyarat Indonesia (BISINDO) merupakan alat komunikasi utama bagi penyandang disabilitas Tuli, namun aplikasi *video call* yang mengintegrasikan pemrosesan AI untuk mendukung komunikasi antara pengguna Tuli dan non-Tuli masih sangat terbatas. Salah satu faktor utama

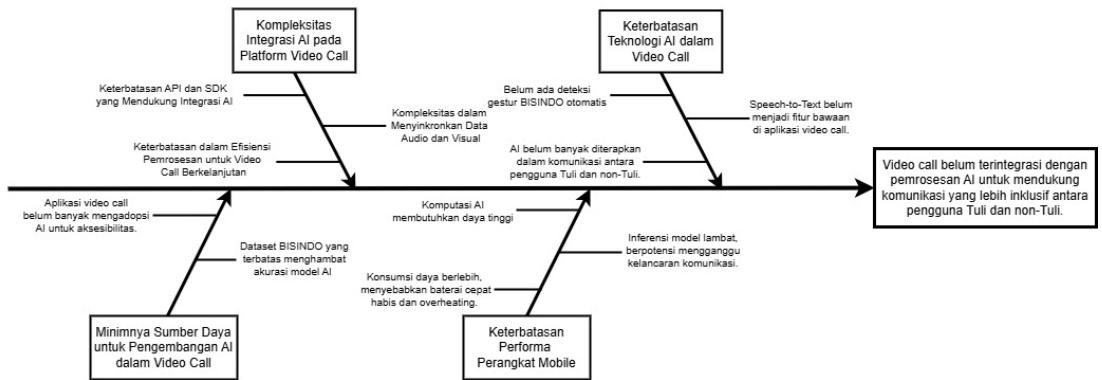
yang menghambat integrasi ini adalah kompleksitas integrasi AI pada platform *video call*. Aplikasi *video call* membutuhkan pemrosesan data visual dan audio secara bersamaan dengan latensi rendah, namun keterbatasan API dan SDK yang mendukung AI membuat penggabungan deteksi gestur BISINDO dan transkripsi suara menjadi lebih rumit. Oleh karena itu, aplikasi *video call* belum mampu mengakomodasi kedua teknologi tersebut secara efektif.

Selain itu, keterbatasan teknologi AI dalam *video call* juga menjadi penghambat utama. Teknologi *speech-to-text* yang diharapkan untuk mengonversi suara pengguna menjadi teks masih belum sepenuhnya terintegrasi dengan aplikasi *video call*. Di sisi lain, deteksi gestur BISINDO yang otomatis juga belum optimal. Hal ini menghambat komunikasi antara pengguna Tuli dan non-Tuli, karena sistem belum dapat mendeteksi seluruh gestur BISINDO dengan akurasi yang tinggi secara *real-time*. Teknologi yang ada saat ini belum mampu menyediakan solusi komunikasi dua arah yang lancar, yang sangat dibutuhkan untuk interaksi inklusif.

Keterbatasan lainnya adalah performa perangkat *mobile* dan minimnya sumber daya untuk pengembangan AI. Pemrosesan model *deep learning* yang membutuhkan daya komputasi tinggi menjadi masalah pada perangkat *mobile* dengan sumber daya terbatas, yang menyebabkan konsumsi daya berlebih dan pemanasan perangkat yang berlebihan. Selain itu, minimnya sumber daya untuk pengembangan AI dalam *video call*, seperti aplikasi *video call* yang belum banyak mengadopsi AI untuk aksesibilitas, menghambat penerapan teknologi yang lebih inklusif. Dataset BISINDO yang terbatas juga menghambat akurasi model AI, karena model yang dilatih dengan dataset yang kecil mungkin tidak mampu menangani variasi gestur secara efektif. Untuk mengatasi tantangan ini, diperlukan integrasi MobileNetV2 yang efisien untuk pendekripsi gestur BISINDO serta *Speech-to-Text* API untuk mentranskripsi suara secara akurat, yang dapat meningkatkan aksesibilitas aplikasi *video call* tanpa mengorbankan

performa

perangkat.



Gambar 3.1 Diagram Ishikawa

3.1.2 Analisis Data

Pada penelitian ini, analisis data berfokus pada pengintegrasian model *deep learning* yang telah dibuat dan *speech recognition* ke dalam aplikasi *video call* ElCue untuk mendeteksi Bahasa Isyarat Indonesia (BISINDO) secara *real-time*. Model *deep learning* telah dikembangkan menggunakan dataset gambar berformat JPEG dengan resolusi 128×128 piksel. Analisis data akan difokuskan pada bagaimana data digunakan dalam proses pendekripsi isyarat dan pengenalan suara saat *video call* berlangsung.

- Data Visual

Data visual dalam sistem ini diperoleh melalui kamera pengguna yang menangkap gerakan tangan dalam aplikasi *video call*. Data yang digunakan meliputi:

1. Input video *real-time*, yang dikirimkan dalam bentuk *frame* per detik (fps) dari kamera pengguna Tuli.
2. Koordinat tangan dan jari, yang diekstraksi dari video untuk dianalisis oleh model *pre-trained* MobileNetV2 guna mengenali gestur BISINDO.
3. Klasifikasi gestur oleh model *deep learning*, yang menghasilkan output teks berdasarkan gestur yang terdeteksi.

Proses ini dilakukan secara *real-time*, di mana sistem menangkap input dari kamera pengguna, memprosesnya melalui model *deep*

learning, dan menampilkan terjemahan dalam bentuk teks kepada pengguna non-Tuli.

- Data Audio

Dalam sistem ini, pengenalan ucapan dilakukan menggunakan *Speech-to-Text* API dari Flutter, yaitu `speech_to_text`. Package ini memungkinkan aplikasi Flutter untuk melakukan pengenalan suara secara *online* dengan bantuan API pengenalan suara bawaan perangkat (seperti Google *Speech* API di Android dan Apple *Speech* API di iOS). API ini memungkinkan konversi suara pengguna non-Tuli menjadi teks secara *real-time* tanpa memerlukan pelatihan model tambahan.

Data yang digunakan dalam proses ini meliputi:

- Input audio *real-time* dari mikrofon pengguna non-Tuli selama sesi *video call*, yang secara otomatis diproses oleh layanan *Speech Recognition* pada perangkat.
- Format audio yang digunakan oleh API, yaitu PCM (Linear16 - WAV), AMR, atau Opus, tergantung pada implementasi layanan *Speech Recognition* di perangkat pengguna.
- Hasil transkripsi teks dari *Speech-to-Text* API, yang diterjemahkan dari suara dan ditampilkan pada layar pengguna Tuli dalam bentuk teks agar dapat dibaca dalam percakapan.

Proses ini memastikan bahwa ucapan pengguna non-Tuli dapat dikonversi menjadi teks secara akurat, sehingga komunikasi dalam *video call* dapat berlangsung dengan lancar, cepat, dan sinkron antara pengguna Tuli dan non-Tuli.

3.1.3 Analisis Kebutuhan

a. Kebutuhan Fungsional

- Sistem harus mampu mendeteksi Bahasa Isyarat Indonesia (BISINDO) menggunakan model *deep learning* berbasis MobileNetV2 yang berjalan langsung di perangkat *mobile* (*on-device inference*).

- Kamera perangkat harus menangkap gerakan tangan pengguna Tuli dan memprosesnya dalam bentuk *frame* per detik (*fps*) untuk diklasifikasikan oleh MobileNetV2 menjadi teks yang ditampilkan dalam *video call*.
- Sistem harus mendukung pengenalan suara menggunakan *speech-to-text* API yang terintegrasi dengan aplikasi *mobile* agar ucapan pengguna non-Tuli dapat dikonversi menjadi teks secara *real-time* dengan akurasi tinggi.
- Proses inferensi MobileNetV2 dan pengenalan suara harus berjalan secara paralel dalam satu aplikasi tanpa mengganggu stabilitas *video call*.
- Aplikasi harus memungkinkan pengguna memilih mode tampilan, seperti hanya menampilkan hasil deteksi gestur, hanya transkripsi suara, atau keduanya secara bersamaan.

b. Kebutuhan Non-Fungsional

- Aplikasi harus mendukung berbagai spesifikasi perangkat *mobile* dengan kinerja optimal, termasuk perangkat dengan kapasitas pemrosesan rendah.
- Pengenalan suara harus tetap berjalan dengan baik dengan tingkat akurasi yang optimal dalam berbagai kondisi lingkungan.
- Antarmuka pengguna harus dioptimalkan agar tetap responsif saat model *deep learning* dan *speech recognition* bekerja secara simultan dalam aplikasi *mobile*.
- Aplikasi harus mampu menangani beban komputasi secara efisien agar tidak menyebabkan overheating atau konsumsi baterai yang berlebihan.

Dengan kebutuhan ini, integrasi MobileNetV2 dan *speech recognition* dalam aplikasi *mobile ElCue* dapat berjalan secara optimal, memberikan pengalaman komunikasi yang lancar dan responsif bagi pengguna Tuli dan non-Tuli.

c. Analisis Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam penelitian ini mencakup lingkungan pengembangan, alat *debugging*, dan *emulator* untuk pengujian aplikasi. Berikut adalah detail spesifikasinya:

Tabel 3.1 Spesifikasi Perangkat Lunak

Spesifikasi	Keterangan
Sistem Operasi	Windows 10
IDE	Visual Studio Code
SDK	Flutter SDK 3.7.0
Bahasa Pemrograman	Dart
Versi Android Minimal	Android 7.0 (API Level 24) atau lebih tinggi
<i>Emulator & Debugging Tools</i>	Android <i>Emulator</i> , ADB

d. Analisis Kebutuhan Perangkat Keras

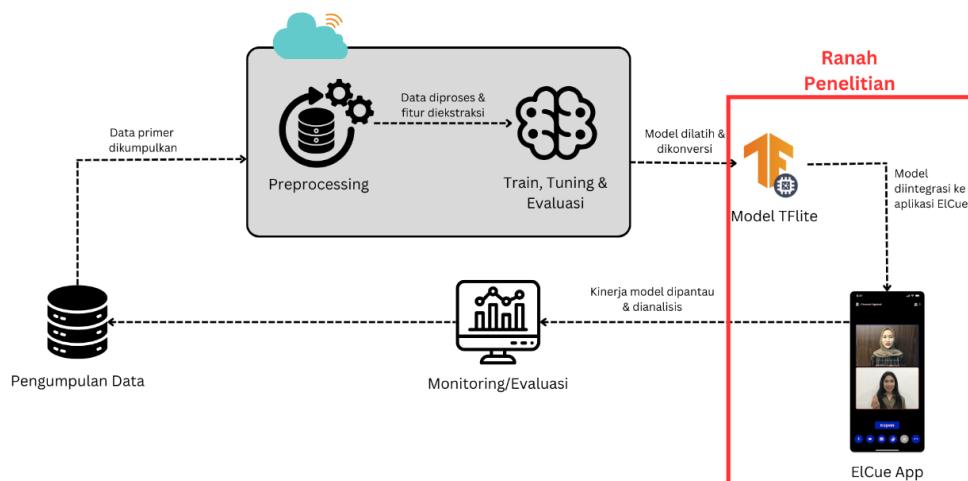
Perangkat keras yang digunakan dalam penelitian ini mencakup spesifikasi minimal dan rekomendasi untuk mendukung proses pengembangan, integrasi model, dan pengujian aplikasi *mobile*. Berikut adalah detail spesifikasinya:

Tabel 3.2 Spesifikasi Perangkat Keras

Spesifikasi	Keterangan
<i>Processor</i>	Intel Core i5-1135G7 (11th Gen) @ 2.40GHz (8 CPUs) atau lebih tinggi
RAM	Minimal 8GB
<i>Storage</i>	SSD 256GB atau lebih
Perangkat Uji Coba	Android 7.0 (API Level 24) atau lebih tinggi
Kamera & Mikrofon	Dibutuhkan untuk pengujian fitur <i>video call</i> dan pendekripsi bahasa

isyarat menggunakan model AI serta
speech recognition

3.1.4 Diagram Umum Sistem



Gambar 3.2 Diagram Umum Sistem

Diagram ini menggambarkan alur sistem dalam pengembangan dan penerapan model deep learning untuk deteksi Bahasa Isyarat Indonesia (BISINDO) dan integrasinya ke dalam aplikasi video call. Proses dimulai dengan *pre-processing data*, di mana dataset gambar berformat JPEG diproses dan fitur diekstraksi untuk melatih model MobileNetV2. Pada tahap *training, tuning & evaluation*, model dilatih dan dievaluasi untuk mendeteksi gestur BISINDO dengan akurasi tinggi. Setelah itu, model dikonversi ke format TFLite dan diintegrasikan dalam aplikasi video call ElCue pada tahap *deployment*, di mana gambar dari video *real-time* diproses dan diklasifikasikan sesuai kosakata BISINDO. Di sisi lain, speech-to-text API digunakan untuk mengonversi ucapan pengguna non-Tuli menjadi teks yang ditampilkan pada layar pengguna Tuli, memungkinkan komunikasi yang efektif tanpa penerjemah bahasa isyarat. Tahap *monitoring/evaluasi* memastikan performa sistem tetap optimal, dengan pemantauan berkelanjutan terhadap penggunaan aplikasi.

Penelitian ini berfokus pada integrasi model *deep learning* ke dalam aplikasi *video call* untuk mendeteksi Bahasa Isyarat Indonesia (BISINDO) secara *real-time*. Model yang digunakan adalah MobileNetV2, yang telah dilatih sebelumnya menggunakan dataset gambar berformat JPEG dengan resolusi 128×128 piksel. MobileNetV2 dipilih karena memiliki arsitektur yang ringan, memungkinkan pemrosesan cepat pada perangkat *mobile* tanpa mengorbankan akurasi.

Integrasi model ke dalam aplikasi dilakukan dengan menangkap *frame* per detik (fps) dari video pengguna Tuli, kemudian mengonversinya menjadi gambar dengan resolusi 128×128 piksel sebelum dikirim ke model untuk diklasifikasikan. Proses ini diawali dengan kamera perangkat yang menangkap video secara *real-time*, lalu *frame* video diambil dalam interval tertentu dan diproses menjadi gambar berformat JPEG. Gambar tersebut kemudian dikirimkan ke model MobileNetV2 yang telah diintegrasikan dalam aplikasi *mobile* untuk diklasifikasikan sesuai dengan 9 kosakata BISINDO, yaitu “A”, “Asal”, “B”, “C”, “D”, “E”, “Halo”, “I”, dan “Terima Kasih”.

Setelah gambar diproses oleh model, hasil klasifikasi berupa teks dari gestur yang terdeteksi akan ditampilkan langsung di layar pengguna non-Tuli dalam aplikasi *video call*. Dengan cara ini, komunikasi antara pengguna Tuli dan non-Tuli dapat berjalan lebih efektif tanpa perlu menggunakan penerjemah bahasa isyarat manual. Model MobileNetV2 yang telah dioptimalkan memastikan bahwa deteksi gestur berjalan dengan cepat dan akurat, serta mampu mengenali variasi gerakan tangan yang berbeda dalam berbagai kondisi pencahayaan dan sudut pandang kamera.

Selain pendekstrian gestur, sistem juga mengintegrasikan *Speech-to-Text* API dari Flutter, yang digunakan untuk menerjemahkan ucapan pengguna non-Tuli menjadi teks yang dapat dibaca oleh pengguna Tuli. Sistem ini bekerja dengan menangkap input audio *real-time* dari mikrofon, yang secara otomatis diproses oleh *Speech Recognition Service* bawaan perangkat, misalnya Google *Speech API* pada Android. Audio yang

dikirimkan ke API ini bergantung pada spesifikasi layanan pengenalan suara yang digunakan, karena setiap platform memiliki standar format audio yang berbeda berdasarkan optimasi untuk akurasi dan efisiensi pemrosesan.

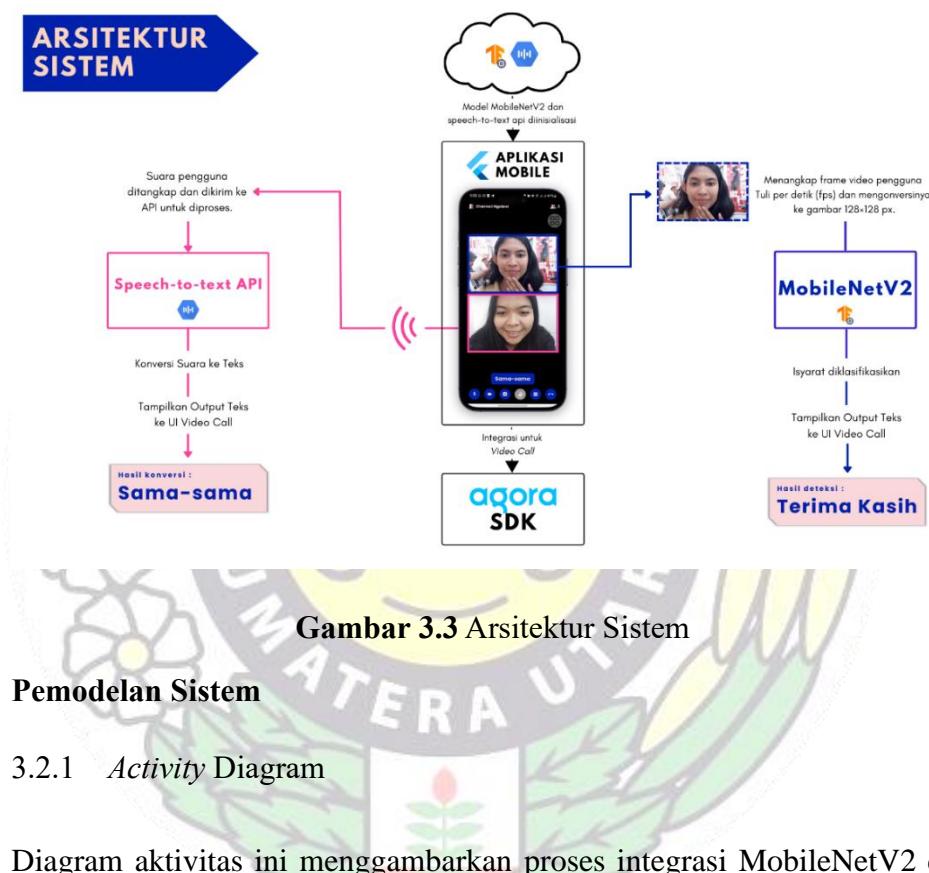
Setelah ucapan dikonversi menjadi teks, hasil transkripsi akan ditampilkan langsung di layar pengguna Tuli, sehingga mereka dapat memahami percakapan dengan lawan bicara mereka tanpa hambatan. Dengan integrasi antara deteksi gestur menggunakan MobileNetV2 dan *Speech-to-Text* API, aplikasi ElCue dapat memfasilitasi komunikasi antara pengguna Tuli dan non-Tuli dalam *video call* tanpa perlu menggunakan alat bantu komunikasi tambahan atau penerjemah bahasa isyarat.

3.1.5 Arsitektur Sistem

Arsitektur sistem dimulai dengan inisialisasi model MobileNetV2 yang sudah diformat dalam TFLite untuk pemrosesan deteksi gestur BISINDO. Model ini diintegrasikan dalam aplikasi untuk mendeteksi gestur tangan pengguna Tuli yang kemudian diterjemahkan menjadi teks dalam *video call*. Bersamaan dengan itu, *Speech-to-Text* API diinisialisasi untuk mengonversi suara yang diucapkan oleh pengguna menjadi teks. Proses ini memungkinkan komunikasi dua arah antara pengguna Tuli dan non-Tuli, di mana suara yang diucapkan oleh non-Tuli akan dikonversi menjadi teks yang ditampilkan di layar *video call*.

Untuk mendukung komunikasi *video call* secara *real-time*, Agora SDK digunakan untuk mengintegrasikan fitur video dan audio *streaming* antara pengguna. Aplikasi menangkap *frame* per detik (fps) dari video pengguna Tuli, yang kemudian diproses dan dikirim ke MobileNetV2 untuk mendeteksi gestur tangan. Setiap gestur yang terdeteksi oleh MobileNetV2, seperti "Terima Kasih", akan diklasifikasikan dan ditampilkan sebagai teks di layar *video call*. Di sisi lain, suara yang ditangkap oleh mikrofon pengguna akan diproses oleh *Speech-to-Text* API, mengonversi ucapan menjadi teks yang kemudian ditampilkan di antarmuka aplikasi *video call*, memudahkan komunikasi dua arah.

Dengan sistem ini, aplikasi *video call* dapat melakukan deteksi gestur BISINDO dan konversi suara menjadi teks secara *real-time*, menyediakan solusi komunikasi yang inklusif dan efektif antara pengguna Tuli dan non-Tuli. Penggunaan MobileNetV2 dan *Speech-to-Text API* yang dioptimalkan dengan Agora SDK memungkinkan integrasi yang lancar antara kedua teknologi, menciptakan pengalaman komunikasi video yang lebih aksesibel.



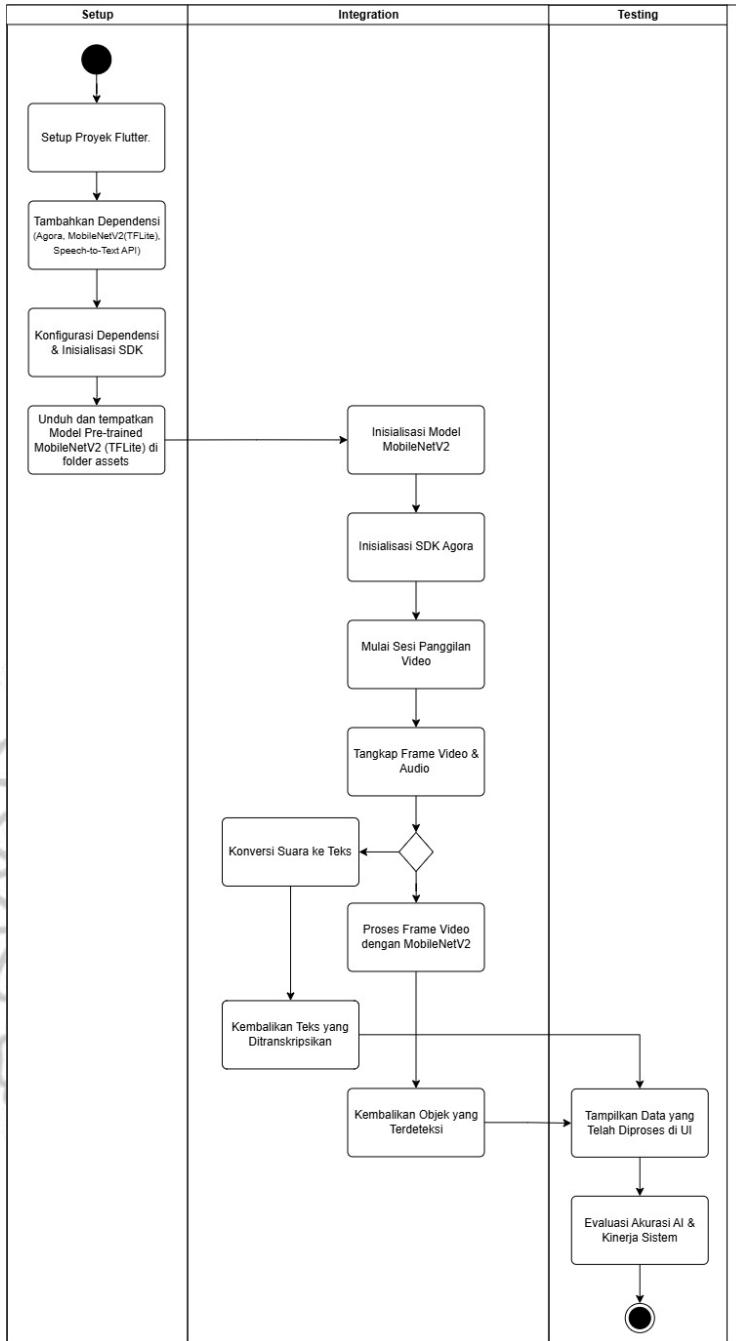
3.2 Pemodelan Sistem

3.2.1 Activity Diagram

Diagram aktivitas ini menggambarkan proses integrasi MobileNetV2 dan *Speech-to-Text API* dalam aplikasi *video call* berbasis Flutter menggunakan Agora SDK. Proses ini terdiri dari tiga tahap utama, yaitu *Setup*, *Integration*, dan *Testing*. Pada tahap *Setup*, proyek Flutter disiapkan dengan menambahkan dependensi yang diperlukan, termasuk Agora SDK untuk komunikasi video, MobileNetV2 dalam format TensorFlow Lite (TFLite) untuk pemrosesan video, dan *Speech-to-Text API* untuk transkripsi suara. Setelah itu, SDK dan API dikonfigurasi agar dapat digunakan dalam sistem. Model MobileNetV2 diunduh dan ditempatkan dalam folder `/assets/` sebagai bagian dari persiapan sebelum digunakan dalam proses inferensi.

Pada tahap *Integration*, model MobileNetV2 diinisialisasi terlebih dahulu sebelum Agora SDK diaktifkan untuk menangani sesi *video call*. Setelah panggilan video dimulai, sistem menangkap *stream* video dan audio untuk diproses lebih lanjut. Proses ini memiliki dua cabang utama, yaitu pemrosesan video dengan MobileNetV2 dan konversi suara dengan *Speech-to-Text API*. Jika data yang diterima berupa *frame* video, MobileNetV2 akan melakukan deteksi objek dan mengembalikan hasil analisisnya. Sementara itu, jika data berupa audio, *Speech-to-Text API* akan mengonversinya menjadi teks yang dapat ditampilkan. Hasil dari kedua proses ini kemudian ditampilkan dalam antarmuka pengguna (UI) untuk memberikan informasi berupa deteksi objek dan teks transkripsi suara dalam *video call*.

Pada tahap *Testing*, evaluasi dilakukan untuk memastikan bahwa sistem berfungsi secara optimal. Pengujian meliputi validasi hasil deteksi objek oleh MobileNetV2, akurasi transkripsi suara yang dihasilkan oleh *Speech-to-Text API*, serta kestabilan performa selama sesi *video call* berlangsung. Jika terdapat kesalahan atau ketidaksesuaian hasil, proses *debugging* dan optimasi dilakukan sebelum sistem diuji ulang. Tahap terakhir dalam diagram ini adalah "Evaluasi Akurasi AI & Kinerja Sistem", yang memastikan bahwa seluruh integrasi berjalan sesuai dengan yang diharapkan. Setelah semua pengujian berhasil, sistem siap digunakan dengan fitur *video call* berbasis kecerdasan buatan yang mampu mendeteksi objek secara *real-time* serta mengubah suara menjadi teks.



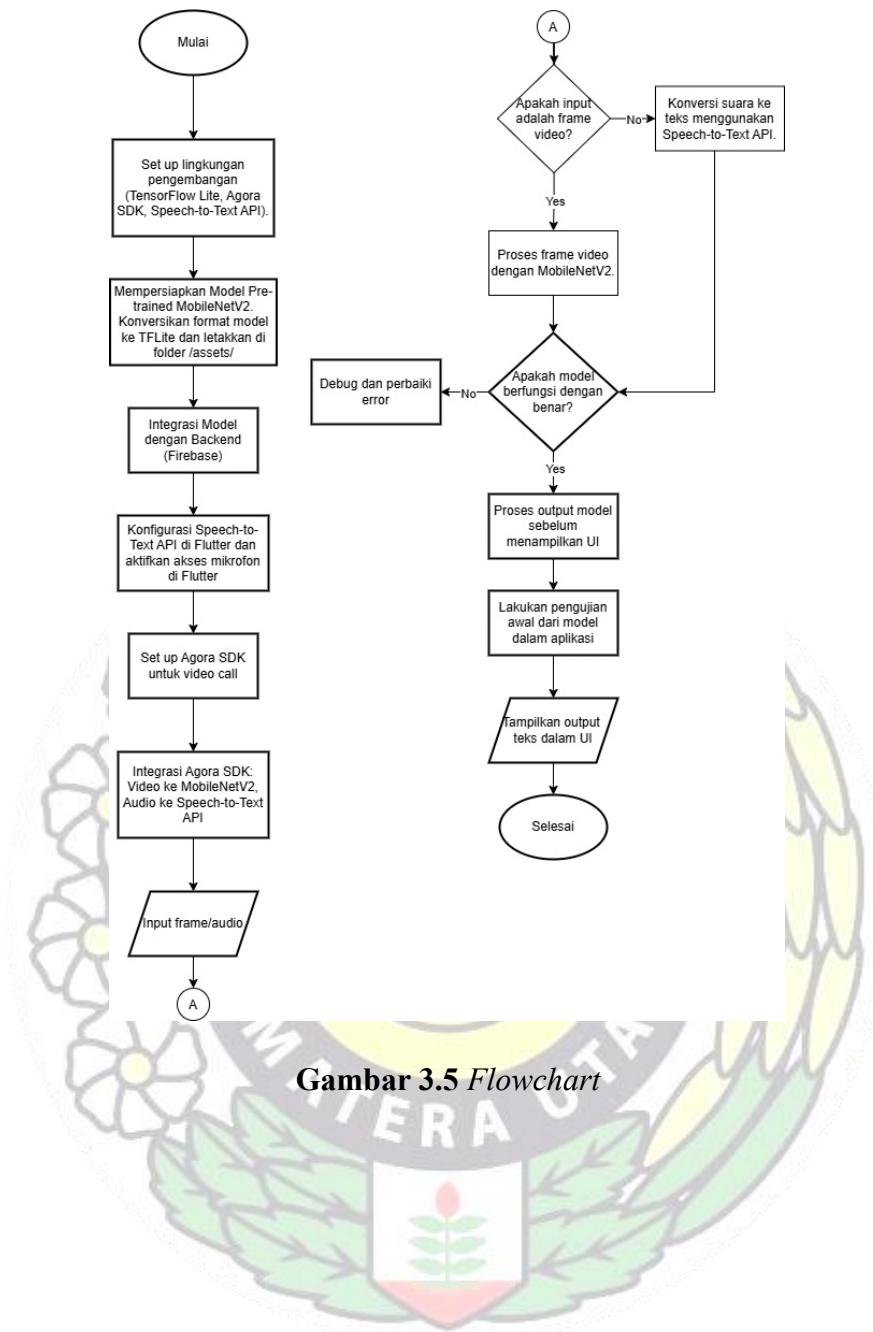
Gambar 3.4 Activity Diagram

3.2.2 Flowchart

Flowchart ini merepresentasikan alur integrasi model *deep learning* MobileNetV2 berbasis TFLite dan *Speech-to-Text API* dalam aplikasi *video call* yang dikembangkan menggunakan Flutter. Proses dimulai dengan inisialisasi *environment* pengembangan, termasuk konfigurasi TensorFlow

Lite untuk inferensi model di perangkat *mobile*, Firebase sebagai *backend* untuk menyimpan data hasil inferensi, Agora SDK untuk menangani komunikasi video dan audio, serta *Speech-to-Text API* untuk menangkap dan mengonversi input suara menjadi teks. Setelah *environment* siap, model MobileNetV2 yang telah dilatih sebelumnya dikonversi ke format TFLite dan disimpan dalam direktori `/assets/`. *Backend* Firebase dikonfigurasi untuk menangani *logging* dan sinkronisasi data, sementara *Speech-to-Text API* dikonfigurasi untuk menerima input suara dengan mengaktifkan akses mikrofon. Agora SDK diintegrasikan agar video dapat diproses oleh MobileNetV2 dan audio dapat dikirim ke *Speech-to-Text API*.

Pada saat *runtime*, sistem menangkap input berupa video *frame* atau audio. Jika input yang diterima adalah video *frame*, data tersebut diproses menggunakan MobileNetV2 untuk melakukan inferensi dan mengidentifikasi gestur atau objek yang relevan. Jika input yang diterima berupa audio, maka sistem akan mengonversinya menjadi teks menggunakan *Speech-to-Text API*. Setelah proses inferensi selesai, sistem melakukan validasi terhadap hasil model. Jika model tidak bekerja dengan benar, sistem akan masuk ke tahap *debugging* dan perbaikan error, kemudian mengulangi proses validasi hingga hasil yang diperoleh sesuai dengan ekspektasi. Jika model berfungsi dengan baik, hasil inferensi diproses lebih lanjut sebelum ditampilkan di UI aplikasi. Setelah dilakukan pengujian akhir, output yang dihasilkan oleh model—baik berupa deteksi gestur dari MobileNetV2 maupun teks transkripsi dari *Speech-to-Text API*—akan ditampilkan kepada pengguna sebelum sistem menyelesaikan eksekusinya.



Gambar 3.5 Flowchart

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Bagian ini membahas penerapan MobileNetV2 untuk deteksi gestur dan *Speech-to-Text* API untuk transkripsi suara ke dalam aplikasi *video call* ElCue yang dikembangkan menggunakan Flutter dan Agora SDK.

4.1.1 Inisialisasi *Environment* Pengembangan

Langkah pertama dalam proses ini adalah setup *environment* untuk memastikan semua *dependency* dan *framework* yang dibutuhkan tersedia dan dikonfigurasi dengan benar.

a. *Framework & Library* yang Digunakan

- Flutter

Flutter digunakan sebagai *framework* utama karena mendukung *asynchronous processing*, memungkinkan MobileNetV2 dan *Speech-to-Text* API berjalan secara *real-time* tanpa menghambat UI *thread*. Dengan ekosistem *plugin* yang luas, integrasi TensorFlow Lite untuk pendekripsi gestur, *Speech-to-Text* API untuk transkripsi suara, dan Agora SDK untuk video *streaming* dapat dilakukan dalam satu aplikasi. Selain itu, fitur *hot reload* mempermudah pengujian dan optimasi sistem tanpa perlu *rebuild* aplikasi secara penuh. Dengan keunggulan ini, Flutter menjadi pilihan yang efisien untuk mengintegrasikan MobileNetV2 dan *Speech-to-Text* API ke dalam aplikasi *video call* berbasis Android.

- Implementasi TensorFlow Lite untuk MobileNetV2

TensorFlow Lite (TFLite) digunakan sebagai *framework* inferensi untuk menjalankan model MobileNetV2 pada perangkat Android. TFLite dipilih karena memiliki performa

yang ringan dan cepat, serta dirancang khusus untuk optimasi pada perangkat *mobile*. Untuk mengintegrasikan TFLite dalam proyek Flutter, *package* `tflite_v2: ^1.0.0` digunakan untuk menjalankan TensorFlow Lite di aplikasi Flutter. Package ini memungkinkan model TFLite yang telah dikonversi digunakan untuk inferensi langsung di perangkat *mobile*. Dengan menggunakan *package* ini, model *deep learning* dapat dipanggil dan dijalankan secara lokal tanpa ketergantungan pada server eksternal, sehingga proses inferensi dapat dilakukan dengan latensi rendah dan efisiensi tinggi. Untuk mengintegrasikan TFLite dalam proyek Flutter, tambahkan *dependencies* TFLite Flutter ke dalam proyek:

```
dependencies:  
    tflite_v2: ^1.0.0
```

Gambar 4.1 *Dependencies* Implementasi TFLite

Kemudian, impor *library* dalam kode Flutter:

```
import 'package:tflite_v2/tflite_v2.dart';
```

Gambar 4.2 *Library* Implementasi TFLite

- Implementasi Agora SDK untuk Ekstraksi *Frame* Video Secara *Real-Time*

Untuk mengimplementasikan Agora SDK dalam ekstraksi *frame* video secara *real-time*, beberapa *dependencies* perlu ditambahkan dalam proyek Flutter. Agora SDK (*agora_rtc_engine*) digunakan untuk menangani komunikasi *video call* dan menangkap *stream* video serta audio secara langsung. Selain itu, *Camera Plugin* (*camera*) digunakan untuk mengakses kamera perangkat dan mengambil *frame* secara independen atau bersamaan dengan Agora SDK sebelum diproses menggunakan MobileNetV2.

Untuk pemrosesan gambar lebih lanjut, *library image* memungkinkan manipulasi dan *pre-processing frame* sebelum dikirim ke model. Dengan kombinasi *dependencies* ini, aplikasi dapat menangkap, memproses, dan mengoptimalkan *frame* video dari sesi *video call* untuk digunakan dalam deteksi gestur secara *real-time* menggunakan MobileNetV2. *Dependencies* yang diperlukan:

```
dependencies:
  agora_rtc_engine: ^5.3.1
  camera: ^0.11.0+1
  image: ^4.2.0
```

Gambar 4.3 *Dependencies* Implementasi Agora SDK

Impor *library* dalam kode flutter:

```
import
'package:agora_rtc_engine/rtc_engine.dart';
import
'package:agora_rtc_engine/rtc_local_view.dart'
as rtc_local_view;
import
'package:agora_rtc_engine/rtc_remote_view.dart'
as rtc_remote_view;
import 'package:camera/camera.dart';
import 'package:image/image.dart';
```

Gambar 4.4 *Library* Implementasi Agora SDK

- Implementasi *Speech-to-Text API*

Untuk mengimplementasikan fitur *Speech-to-Text* dalam proyek Flutter, beberapa *dependencies* telah ditambahkan dalam file *pubspec.yaml*. *library speech_to_text* digunakan sebagai solusi utama untuk menangani konversi suara menjadi teks secara *real-time* dalam aplikasi. Selain

itu, `flutter_speech` dapat digunakan sebagai alternatif dalam mengolah input suara dengan dukungan berbagai mesin pengenalan suara. Jika diperlukan akurasi tinggi dengan pemrosesan berbasis *cloud*, `google_speech` memungkinkan integrasi dengan Google *Speech-to-Text* API, yang memerlukan konfigurasi API Key dari Google Cloud. Dengan kombinasi *dependencies* ini, aplikasi dapat menangkap input audio pengguna, mengonversinya menjadi teks, dan menampilkan hasil transkripsi dalam UI secara dinamis selama sesi *video call* berlangsung.

```
dependencies:
  speech_to_text: ^6.6.0
  flutter_speech: ^2.0.0
  google_speech: ^5.2.0+1
```

Gambar 4.5 Dependencies Implementasi Speech-to-Text API

Impor *library* dalam kode flutter:

```
import
'package:speech_to_text/speech_to_text.dart'
as stt;
import
'package:flutter_speech/flutter_speech.dart';
import
'package:google_speech/google_speech.dart';
```

Gambar 4.6 Library Implementasi Speech-to-Text API

b. Platform Pengujian: Android

Karena aplikasi ini hanya berjalan di Android, maka pengujian dilakukan pada perangkat dengan spesifikasi berikut:

Tabel 4.1 Spesifikasi Platform Pengujian Android

Spesifikasi	Keterangan
-------------	------------

OS	Android 7.0 (API Level 24) atau lebih tinggi
Processor	Octa-core (Snapdragon 660 / MediaTek Helio G80) atau lebih tinggi
RAM	Minimum 4GB (Direkomendasikan 6GB+)
GPU	Adreno 512 / Mali-G52 atau lebih tinggi
Storage	Minimum 64GB (Direkomendasikan 128GB+)

4.1.2 Implementasi MobileNetV2 untuk Pendekripsi BISINDO

Implementasi MobileNetV2 dalam aplikasi bertujuan untuk mendekripsi gestur BISINDO secara *real-time* selama sesi *video call*. Proses ini mencakup pengambilan *frame* dari *video call*, prapemrosesan gambar sebelum inferensi, klasifikasi menggunakan MobileNetV2, serta penampilan hasil deteksi sebagai *overlay* pada UI.

a. Pemuatan dan Inferensi MobileNetV2 (TFLite)

Kode ini merupakan implementasi dalam bahasa Dart yang menggunakan pustaka TensorFlow Lite (TFLite) untuk mendekripsi bahasa isyarat dari gambar. Kelas utama, `SignLanguageModel`, memiliki tiga fungsi utama. Fungsi pertama, `loadModel()`, bertugas untuk memuat model pembelajaran mesin yang telah dilatih sebelumnya dalam format TensorFlow Lite (`model3.tflite`) serta file labelnya (`label3.txt`). Jika pemuatan model berhasil, pesan keberhasilan akan dicetak ke konsol, dan jika gagal, pesan kesalahan akan ditampilkan. Fungsi kedua, `detectSignLanguage(File image)`, digunakan untuk menjalankan model yang telah dimuat pada sebuah gambar yang diberikan. Proses ini melibatkan normalisasi nilai piksel dan menetapkan batasan kepercayaan minimal sebesar 50% untuk hasil klasifikasi. Jika model berhasil mengidentifikasi bahasa isyarat dalam gambar, maka hasil pengenalan akan dikembalikan dalam bentuk daftar objek. Jika terjadi kesalahan, pesan kesalahan akan dicetak ke konsol. Fungsi

terakhir, `dispose()`, berfungsi untuk menutup model dan melepaskan sumber daya yang digunakan oleh TFLite setelah proses klasifikasi selesai guna menghindari kebocoran memori. Dengan demikian, kode ini memungkinkan aplikasi untuk mengenali bahasa isyarat dari gambar menggunakan model *deep learning* yang telah dilatih, langsung di perangkat tanpa perlu koneksi internet.

```
import 'dart:io';
import 'package:tflite_v2/tflite_v2.dart';

class SignLanguageModel {
    Future<void> loadModel() async {
        try {
            String? res = await Tflite.loadModel(
                model: 'assets/model3.tflite',
                labels: 'assets/label3.txt',
            );
            print('Model loaded: $res');
        } catch (e) {
            print('Failed to load model: $e');
        }
    }

    Future<List<dynamic>?> detectSignLanguage(File
image) async {
        try {
```

```

        var recognitions = await
Tflite.runModelOnImage(
    path: image.path,
    imageMean: 0.0,
    imageStd: 255.0,
    numResults: 5,
    threshold: 0.5,
);
return recognitions;
} catch (e) {
print('Failed to run model on image: $e');
return null;
}
}

void dispose() {
Tflite.close();
}
}

```

Gambar 4.7 Code untuk Pemuatan dan Inferensi MobileNetV2 (TFLite)

b. Pengambilan *Frame* Video dari Agora SDK

Pada bagian pertama, pengambilan *frame* video dari Agora SDK dilakukan melalui *event handler* `userJoined`, yang dipanggil ketika pengguna baru bergabung ke dalam panggilan video. Di dalam fungsi ini, objek `AgoraUser` dibuat untuk mewakili pengguna yang bergabung, dan tampilan videonya ditampilkan menggunakan `rtc_remote_view.SurfaceView`, yang merupakan *widget* dari Agora SDK untuk menampilkan video pengguna jarak jauh. Untuk memungkinkan pengambilan gambar dari tampilan video ini, *widget* `rtc_remote_view.SurfaceView` dibungkus dalam *widget* `Screenshot`, yang dikontrol oleh `screenshotController`. Dengan cara ini, setiap kali video

pengguna jarak jauh ditampilkan, sistem juga dapat menangkap tangkapan layar dari tampilan tersebut untuk digunakan dalam proses analisis lebih lanjut, seperti deteksi bahasa isyarat.

```

userJoined: (uid, elapsed) {
    final info = 'LOG::userJoined: $uid';
    debugPrint(info);
    setState(() {
        _remoteUid = uid;
        _users.add(
            AgoraUser(
                uid: uid,
                view: Screenshot(
                    controller: screenshotController,
                    child: rtc_remote_view.SurfaceView(
                        channelId: widget.channelName,
                        uid: uid,
                    ),
                ),
            ),
        );
    });
},

```

Gambar 4.8 Code untuk Pengambilan Frame Video Setiap Pengguna Baru Bergabung

Pada bagian kedua, pengambilan *frame* dari *video call* dilakukan secara berkala menggunakan fungsi `_startTakingScreenshots()`, yang menginisialisasi timer (`_ScreenshotTimer`) dengan interval 1 detik untuk menjalankan proses pengambilan tangkapan layar. Jika terdapat pengguna jarak jauh (`_remoteUid != null`), fungsi `screenshotController.capture()` dipanggil untuk mengambil tangkapan layar dari tampilan video yang sebelumnya sudah dibungkus dengan *widget* `Screenshot`. Jika gambar berhasil diambil, gambar tersebut dikirim ke fungsi

`_detectSignLanguage(image)`, yang akan menganalisis gambar tersebut menggunakan model *deep learning* untuk mengenali bahasa isyarat. Dengan pendekatan ini, sistem dapat secara otomatis menangkap *frame* video dari pengguna jarak jauh dan memprosesnya untuk deteksi bahasa isyarat tanpa intervensi manual.

```
void _startTakingScreenshots() {
    _ScreenshotTimer?.cancel();
    _ScreenshotTimer = Timer.periodic(Duration(seconds: 1), (timer) {
        if (_remoteUid != null) {

            screenshotController.capture().then((Uint8List?
            image) {
                if (image != null) {
                    _detectSignLanguage(image);
                }
            }).catchError((onError) {
                print(onError);
            });
        }
    });
}
```

Gambar 4.9 Code untuk Pengambilan *Frame Video Call* Secara Berkala

c. Prapemrosesan Gambar

Setelah *frame* video diperoleh, langkah selanjutnya adalah prapemrosesan agar sesuai dengan input yang dibutuhkan oleh MobileNetV2. Proses prapemrosesan gambar dimulai dengan mendapatkan direktori sementara menggunakan `getTemporaryDirectory()`, yang berfungsi sebagai lokasi penyimpanan file gambar hasil *screenshot*. Setelah itu, variabel `path` dibuat dengan menentukan lokasi file gambar dalam direktori

tersebut dengan nama `frame.jpg`. Selanjutnya, gambar yang masih dalam bentuk `Uint8List` dikonversi dan disimpan sebagai file `.jpg` menggunakan `File(path).writeAsBytes(imageBytes)`. Langkah ini memastikan bahwa gambar yang diambil dari *screenshot* diubah menjadi format yang dapat dibaca oleh model sebelum dilakukan proses deteksi bahasa isyarat.

```
final directory = await getTemporaryDirectory();
final path = '${directory.path}/frame.jpg';
await File(path).writeAsBytes(imageBytes);
```

Gambar 4.10 *Code* untuk Prapemrosesan Gambar

d. Inferensi dengan MobileNetV2

Proses inferensi dengan model MobileNetV2 dalam kode ini dilakukan melalui fungsi `_detectSignLanguage(Uint8List imageBytes)`, yang bertanggung jawab untuk mengenali bahasa isyarat dari *frame* video yang diambil. Saat sebuah *frame* ditangkap dari *video call*, data gambar disimpan dalam format `Uint8List`, lalu dikonversi menjadi file gambar sementara (`frame.jpg`) di dalam direktori penyimpanan sementara perangkat menggunakan `getTemporaryDirectory()`. Setelah gambar tersimpan, fungsi ini memanggil `_signLanguageModel.detectSignLanguage(File(path))`, yang merupakan metode dalam kelas `SignLanguageModel` yang telah dikaitkan dengan model *deep learning* berbasis MobileNetV2 yang dikonversi ke format TensorFlow Lite (TFLite). Model ini kemudian menganalisis gambar dan mengembalikan hasil klasifikasinya dalam bentuk daftar label yang sesuai dengan bahasa isyarat yang dikenali. Jika hasil inferensi berhasil dan model mendeteksi suatu bahasa isyarat, teks hasil klasifikasi akan diperbarui dalam variabel `_signLanguageText` menggunakan `setState()`, sehingga hasilnya dapat langsung ditampilkan di

antarmuka pengguna. Dengan cara ini, sistem dapat secara otomatis mendekripsi bahasa isyarat dari *video call* secara *real-time*, memungkinkan komunikasi yang lebih mudah antara pengguna yang menggunakan BISINDO dan lawan bicaranya.

```
void _detectSignLanguage(Uint8List imageBytes)
async {
    final directory = await
getTemporaryDirectory();
    final path = '${directory.path}/frame.jpg';
    await File(path).writeAsBytes(imageBytes);

    final result = await
_signLanguageModel.detectSignLanguage(File(path));
    if (result != null && result.isNotEmpty) {
        setState(() {
            _signLanguageText = result[0]['label'];
        });
    }
}
```

Gambar 4.11 Code untuk Inferensi dengan MobileNetV2

- Menampilkan Teks Hasil Deteksi di UI *Video Call*

Pada kode di bawah, bagian yang bertanggung jawab untuk menampilkan teks hasil deteksi bahasa isyarat dalam UI *video call* terdapat pada *widget Visibility*. Kode ini memastikan bahwa teks hanya akan ditampilkan jika variabel `_isSignLanguageTextVisible` bernilai `true`. Teks hasil deteksi ditampilkan dalam sebuah *Container* yang memiliki padding dan dekorasi khusus, termasuk warna latar belakang biru (`Color(0xFF0021AC)`) serta sudut yang membulat (`borderRadius: BorderRadius.circular(8.0)`). Teks yang ditampilkan diperoleh dari variabel `_signLanguageText`, yang sebelumnya diperbarui dalam fungsi

`_detectSignLanguage(Uint8List imageBytes)`. Proses pembaruan dilakukan dengan `setState()`, sehingga UI dapat menampilkan hasil deteksi secara *real-time* setiap kali model mengenali bahasa isyarat dari gambar yang diproses. Selain itu, teks diberikan ukuran font sebesar 23 dan warna teks `Color(0xFFFAD7DB)`, agar terlihat kontras dengan latar belakangnya. Dengan mekanisme ini, pengguna dapat melihat hasil klasifikasi bahasa isyarat secara langsung selama *video call* berlangsung.

```
Visibility(
    visible: _isSignLanguageTextVisible,
    child: Container(
        padding: EdgeInsets.symmetric(vertical: 8.0,
horizontal: 16.0),
        decoration: BoxDecoration(
            color: Color(0xFF0021AC),
            borderRadius: BorderRadius.circular(8.0),
        ),
        child: Text(
            '_signLanguageText',
            style: TextStyle(color: Color(0xFFFAD7DB),
fontSize: 23),
        ),
    ),
),
```

Gambar 4.12 *Code untuk Menampilkan Teks Hasil Deteksi di UI Video Call*

4.1.3 Integrasi *Speech-to-Text* API untuk Pengenalan Suara

a. Menginisialisasi *Speech-to-Text*

Pada kode di bawah, inisialisasi *Speech-to-Text* dilakukan melalui metode `_initSpeechToText()`. Di dalam metode ini, objek `SpeechToText` diinisialisasi dengan memanggil `initialize()`

yang akan mengonfigurasi pengenalan suara. Callback `onStatus` memantau perubahan status pengenalan suara, dan `onError` menangani potensi kesalahan. Setelah berhasil diinisialisasi, sistem akan mencetak apakah pengenalan suara tersedia atau tidak. Metode ini dipanggil dalam `initState()` untuk memastikan fitur *speech-to-text* siap digunakan begitu aplikasi dimulai.

```
Future<void> _initSpeechToText() async {
    bool available = await _speech.initialize(
        onStatus: (status) {
            print('onStatus: $status');
            if (status == 'notListening') {
                setState(() {
                    _transcription = '';
                });
            }
        },
        onError: (val) {
            print('onError: $val');
        },
    );
    if (available) {
        print('Speech recognition available');
    } else {
        print('Speech recognition not available');
    }
}
```

Gambar 4.13 Code untuk Menginisialisasi *Speech-to-Text*

b. Mendengarkan Suara dan Mengonversi ke Teks

Proses untuk mendengarkan suara dan mengonversinya ke teks dilakukan melalui objek `SpeechToText` yang diinisialisasi dalam metode `_initSpeechToText()`. Kemudian, metode `_startListening()` digunakan untuk mulai mendengarkan suara dengan memanggil fungsi `listen()`. Hasil pengenalan suara dikirim melalui callback `onResult`, di mana teks yang dikenali

disimpan dalam variabel `_transcription`. Variabel ini kemudian diperbarui menggunakan `setState()` untuk menampilkan teks transkripsi secara *real-time*. Dengan pengaturan `localeId: 'id_ID'`, aplikasi dikonfigurasi untuk mengenali suara dalam bahasa Indonesia, memungkinkan pengguna untuk berbicara dan melihat teks yang dihasilkan dari percakapan mereka.

```
void _startListening() {
    _speech.listen(
        onResult: (val) {
            setState(() {
                _transcription = val.recognizedWords;
            });
            _resetTranscriptionTimer();
        },
        localeId: 'id_ID',
    );
}
```

Gambar 4.14 Code untuk Mendengarkan Suara dan Mengonversi ke Teks

c. Menampilkan Transkripsi di Layar

Bagian yang bertanggung jawab untuk menampilkan transkripsi di layar adalah bagian yang menggunakan `widget Visibility` dan `Container` untuk menampilkan teks transkripsi yang disimpan dalam variabel `_transcription`. Teks transkripsi ini akan ditampilkan hanya jika fitur transkripsi diaktifkan (ketika `_isTranscriptionEnabled bernilai true`). Widget `Text` di dalam `Container` akan menampilkan teks transkripsi dengan styling tertentu. Setiap kali hasil pengenalan suara diperbarui, variabel `_transcription` akan diubah dan tampilan akan otomatis diperbarui dengan teks baru.

```
Visibility(
    visible: _isTranscriptionEnabled,
    child: Container(
        padding: EdgeInsets.symmetric(vertical: 8.0, horizontal: 16.0),
        decoration: BoxDecoration(
            color: Color(0xFF0021AC),
            borderRadius: BorderRadius.circular(8.0),
        ),
        child: Text(
            '_transcription',
            style: TextStyle(color: Color(0xFFFFAD7DB), fontSize: 23),
        ),
    ),
),
```

Gambar 4.15 Code untuk Menampilkan Transkripsi di Layar

d. Menangani Transkripsi *Timeout*

Penanganan transkripsi *timeout* dilakukan melalui metode `_resetTranscriptionTimer()`. Metode ini berfungsi untuk mengatur jeda waktu atau timeout setelah hasil transkripsi terakhir, di mana teks transkripsi akan dihapus setelah 2 detik jika tidak ada hasil baru. Setiap kali ada hasil transkripsi baru, timer akan direset, dan jika tidak ada hasil dalam waktu 2 detik, maka teks transkripsi akan dihapus dengan mengosongkan variabel `_transcription`. Ini memastikan bahwa teks yang ditampilkan di layar hanya menunjukkan hasil pengenalan suara terbaru dan tidak akan terus tertampilkan setelah percakapan selesai.

```

void _resetTranscriptionTimer() {
    _transcriptionTimer?.cancel();
    _transcriptionTimer = Timer(Duration(seconds: 2), () {
        setState(() {
            _transcription = '';
        });
    });
}

```

Gambar 4.16 *Code* untuk Menangani Transkripsi *Timeout*

e. Memulai dan Menghentikan Pengenalan Suara

Metode `_startListening()` bertugas untuk memulai proses pengenalan suara dengan memanggil fungsi `listen()` pada objek `SpeechToText` yang telah diinisialisasi. Fungsi ini mendengarkan suara pengguna dan mengubahnya menjadi teks yang kemudian disimpan dalam variabel `_transcription`. Metode ini juga mengatur bahasa yang digunakan untuk pengenalan suara, dalam hal ini menggunakan `localeId: 'id_ID'` untuk bahasa Indonesia.

Berikut adalah kode untuk memulai pengenalan suara:

```

void _startListening() {
    _speech.listen(

```

```

        onResult: (val) {
            setState(() {
                _transcription = val.recognizedWords;
            });
            _resetTranscriptionTimer();
        },
        localeId: 'id_ID',
    );
}

void _stopListening() {
    _speech.stop();
}

```

Gambar 4.17 Code untuk Memulai dan Menghentikan Pengenalan Suara

4.2 Pengujian Sistem

Bagian ini menjelaskan metode dan hasil pengujian yang dilakukan untuk mengevaluasi akurasi, kecepatan, dan performa integrasi MobileNetV2 dan *Speech-to-Text API* dalam aplikasi *video call*.

4.2.1 Hasil Pengujian Pendekripsi Gestur

Pengujian dilakukan dengan 6 kelas gestur BISINDO yang berbeda, yang masing-masing diuji dalam 20 percobaan. Setiap percobaan melibatkan pengguna yang melakukan gestur yang sesuai dengan kosakata BISINDO yang telah ditentukan, dan sistem diminta untuk mendeteksi gerakan tersebut dalam aplikasi *video call* secara *real-time*. Untuk menghitung akurasi dari sistem, digunakan rumus berikut:

$$\text{Akurasi} = \frac{\text{Jumlah Deteksi Benar}}{\text{Jumlah Total Percobaan}} \times 100\%$$

Dengan menggunakan rumus ini, tingkat akurasi sistem dalam mendekripsi gestur yang dilakukan oleh pengguna dapat dievaluasi. Akurasi

dihitung dengan membandingkan jumlah deteksi yang benar terhadap jumlah total percobaan yang dilakukan selama pengujian.

Tabel 4.2 Hasil Pengujian Pendekripsi Gestur

Kelas Gestur	Jumlah Percobaan	Prediksi Benar	Prediksi Salah	Akurasi (%)
A	20	17	3	85%
Asal	20	15	5	75%
B	20	17	3	85%
C	20	18	2	90%
D	20	19	1	95%
E	20	18	2	90%
Halo	20	17	3	85%
I	20	18	2	90%
Terima kasih	20	15	5	75%
Rata-Rata	180	154	26	85,56%

Setelah model MobileNetV2 diintegrasikan ke dalam aplikasi *video call*, sistem pendekripsi gestur BISINDO berhasil mencapai rata-rata akurasi 85,56%. Hasil ini menunjukkan bahwa model dapat mendekripsi gestur dengan baik dalam kondisi *real-time* meskipun menghadapi tantangan seperti perubahan posisi pengguna, pencahayaan yang tidak konsisten, dan variasi gerakan tangan. Beberapa kelas gestur, seperti kelas D, C, E, dan I, menunjukkan akurasi yang sangat tinggi, mencapai 90% atau lebih. Akurasi yang tinggi ini kemungkinan disebabkan oleh kesederhanaan dan kejelasan gerakan tangan pada kosakata-kosakata tersebut, yang lebih mudah dikenali oleh model.

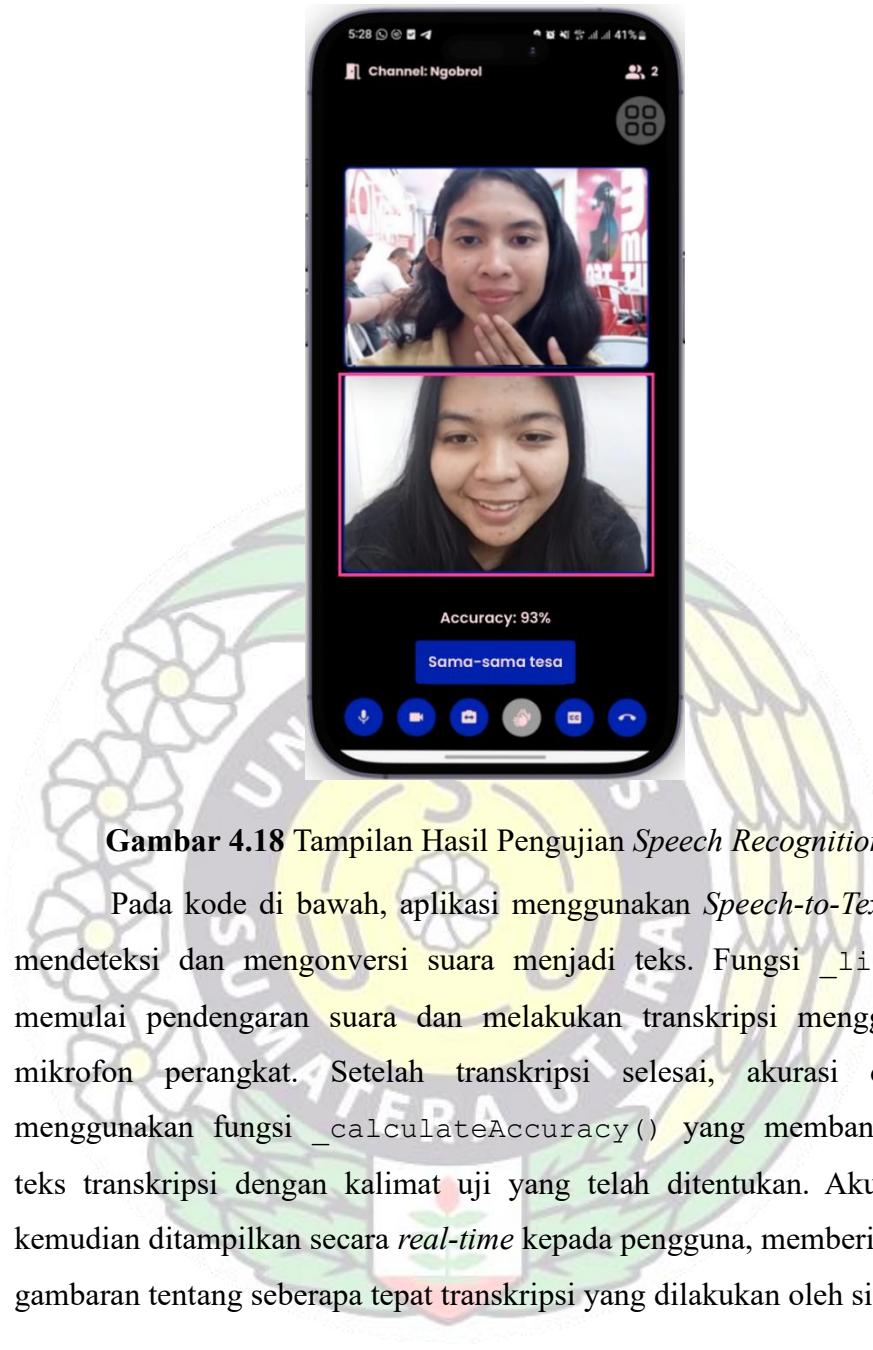
Di sisi lain, beberapa kelas seperti "Asal" dan "Terima Kasih" memiliki akurasi sedikit lebih rendah, yaitu sekitar 75%, yang mungkin disebabkan oleh kerumitan gerakan tangan yang lebih rumit atau posisi tangan yang lebih sulit dikenali oleh model. Variasi dalam cara pengguna

melakukan gestur juga dapat mempengaruhi akurasi pada kelas-kelas tersebut.

Secara keseluruhan, model menunjukkan performa yang memadai untuk aplikasi deteksi gestur dalam *video call*. Akurasi ini mencerminkan keberhasilan integrasi MobileNetV2 dalam aplikasi, meskipun masih ada peluang untuk meningkatkan deteksi gestur lebih lanjut melalui penyesuaian model dan penggunaan dataset yang lebih beragam.

4.2.2 Hasil Pengujian *Speech Recognition*

Dalam pengujian ini, aplikasi yang telah mengintegrasikan *Speech-to-Text* API berhasil mentranskripsi suara pengguna secara langsung dan menampilkan tingkat akurasi transkripsi pada layar *video call* secara *real-time*. Aplikasi menghitung akurasi transkripsi berdasarkan perbandingan antara teks yang diucapkan dan teks yang dihasilkan oleh sistem. Hasil pengujian menunjukkan bahwa aplikasi ini mencapai rata-rata akurasi 93%, yang mencerminkan kemampuan sistem dalam mengenali dan mengonversi ucapan menjadi teks dengan tingkat ketepatan yang cukup tinggi. Akurasi 93% diperoleh karena aplikasi dapat mengenali sebagian besar kata-kata yang diucapkan dengan jelas, terutama dalam kalimat sederhana dan terstruktur. Faktor-faktor seperti kualitas mikrofon, kecepatan bicara yang moderat, serta sedikit gangguan suara latar berkontribusi pada hasil yang sangat baik. Meskipun ada beberapa variasi aksen dan pengucapan, sistem berhasil mengenali kata-kata dengan tingkat kepercayaan yang tinggi, menghasilkan akurasi yang solid dalam kondisi pengujian yang dilakukan. Tampilan hasil akurasi dalam aplikasi hanya dilakukan untuk tujuan pengujian dan dihilangkan dalam implementasi aplikasi yang sesungguhnya.



Gambar 4.18 Tampilan Hasil Pengujian *Speech Recognition*

Pada kode di bawah, aplikasi menggunakan *Speech-to-Text* untuk mendeteksi dan mengonversi suara menjadi teks. Fungsi `_listen()` memulai pendengaran suara dan melakukan transkripsi menggunakan mikrofon perangkat. Setelah transkripsi selesai, akurasi dihitung menggunakan fungsi `_calculateAccuracy()` yang membandingkan teks transkripsi dengan kalimat uji yang telah ditentukan. Akurasi ini kemudian ditampilkan secara *real-time* kepada pengguna, memberi mereka gambaran tentang seberapa tepat transkripsi yang dilakukan oleh sistem.

Dengan pengujian ini, dapat disimpulkan bahwa aplikasi berhasil memberikan transkripsi yang akurat dengan tingkat akurasi rata-rata 93%, menjadikannya efektif untuk percakapan *video call* dengan integrasi *speech recognition*.

```
import 'package:speech_to_text/speech_to_text.dart' as stt;
import 'package:flutter/material.dart';

late stt.SpeechToText _speech;
bool _isListening = false;
String _transcribedText = '';
double _confidence = 0.0;
String _audioText = '';

double _calculateAccuracy(String transcribedText) {
    List<String> transcribedWords =
transcribedText.split(' ');
    int correctCount = transcribedWords.length;

    return (correctCount / transcribedWords.length) * 100;
}

void _listen() async {
    if (!_isListening) {
        bool available = await _speech.initialize(
            onStatus: (val) {
                print('onStatus: $val');
            },
            onError: (val) {
                print('onError: $val');
            },
        );
        if (available) {
            setState(() {
                _isListening = true;
            });
        }
    }
}
```

```

        _speech.listen(
            onResult: (val) {
                setState(() {
                    _transcribedText = val.recognizedWords;
                    if (val.hasConfidenceRating && val.confidence > 0) {
                        _confidence = val.confidence;
                    }
                });

                double accuracy = _calculateAccuracy(_transcribedText);
                print('Akurasi: $accuracy%');
                print('Teks yang diucapkan: ${_transcribedText}');
            });
        },
        localeId: 'id_ID',
        listenFor: Duration(seconds: 30),
        pauseFor: Duration(seconds: 5),
        onSoundLevelChange: (double level) {
            print("Tingkat suara: $level");
        },
        cancelOnError: true,
        listenMode: stt.ListenMode.dictation,
    );
}
} else {
    setState(() {
        _isListening = false;
    });
    _speech.stop();
}
}
}

```

```
@override  
void initState() {  
    super.initState();  
    _speech = stt.SpeechToText();  
}  
  
@override  
void dispose() {  
    _speech.stop();  
    super.dispose();  
}
```

Gambar 4.19 *Code untuk Pengujian Akurasi Speech Recognition*



BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah tahap implementasi dilakukan dalam penelitian integrasi model *deep learning* dengan *pre-trained* MobileNetV2 dan *Speech Recognition* pada aplikasi *video call* ElCue untuk pendekripsi Bahasa Isyarat Indonesia (BISINDO) secara *real-time*, serta melalui pengujian terhadap akurasi model dan performa *real-time*, maka diperoleh kesimpulan sebagai berikut:

1. Penelitian ini berhasil mengintegrasikan MobileNetV2 untuk deteksi gestur BISINDO dan *Speech-to-Text* API untuk konversi suara menjadi teks dalam aplikasi *video call*, memungkinkan komunikasi dua arah antara pengguna Tuli dan non-Tuli.
2. MobileNetV2 berhasil mendeteksi gestur BISINDO dengan akurasi rata-rata 85,56% dalam kondisi *real-time* meskipun terdapat tantangan seperti variasi gerakan tangan dan kondisi pencahayaan yang berubah.
3. *Speech-to-Text* API berhasil mentranskripsi suara pengguna non-Tuli menjadi teks dengan akurasi rata-rata 93%, memungkinkan komunikasi yang lebih lancar dan efektif.
4. Agora SDK diintegrasikan dengan sukses untuk menangani *video call*, memastikan kualitas komunikasi video dan audio tetap lancar, meskipun menggunakan teknologi AI yang memerlukan pemrosesan data *real-time*.
5. Keterbatasan daya komputasi perangkat *mobile* menjadi tantangan utama dalam pemrosesan model *deep learning*, yang mempengaruhi konsumsi daya dan performa perangkat. Optimalisasi lebih lanjut diperlukan agar sistem dapat berjalan lebih efisien di perangkat dengan sumber daya terbatas.

5.2 Saran

Berdasarkan hasil penelitian dan pengujian yang dilakukan, terdapat beberapa saran yang dapat dijadikan bahan pertimbangan untuk pengembangan lebih lanjut:

1. Akurasi model *deep learning* dapat ditingkatkan dengan penggunaan dataset BISINDO yang lebih besar dan bervariasi. Pengumpulan dataset yang lebih banyak dapat membantu model mengenali variasi gerakan tangan dengan lebih baik, serta meningkatkan akurasi pada kelas gestur yang lebih rumit.
2. Meskipun aplikasi berjalan dengan baik, penggunaan perangkat *mobile* dengan sumber daya terbatas dapat menghambat performa sistem. Oleh karena itu, optimasi lebih lanjut diperlukan dalam hal konsumsi daya dan pemrosesan model agar aplikasi tetap berjalan dengan lancar tanpa mempengaruhi kinerja perangkat *mobile* secara signifikan.
3. Meskipun *Speech-to-Text* API berhasil mentranskripsi ucapan dengan akurat, pengenalan suara dengan aksen yang berbeda atau dalam kondisi kebisingan masih memiliki ruang untuk perbaikan. Mengintegrasikan model pengenalan suara berbasis AI yang lebih terlatih dapat membantu meningkatkan akurasi dalam kondisi lingkungan yang kurang ideal.
4. MobileNetV2, teknologi AI lainnya, seperti *Convolutional Neural Networks* (CNN) atau *Recurrent Neural Networks* (RNN), dapat dieksplorasi untuk meningkatkan akurasi deteksi gestur dan transkripsi suara. Pemanfaatan model-model ini dapat memberikan hasil yang lebih akurat dengan memperhitungkan variasi gerakan tangan dan pengucapan yang lebih kompleks.

Dengan implementasi dan perbaikan lebih lanjut, sistem yang telah dikembangkan dalam penelitian ini dapat memberikan kontribusi besar bagi peningkatan aksesibilitas komunikasi antara pengguna Tuli dan non-Tuli, serta membuka peluang bagi pengembangan teknologi komunikasi yang lebih inklusif di masa depan.

DAFTAR PUSTAKA

- Ahmad Azzam Alhanafi, & Arrie Kurniawardhani. (2024). Object Detection for Retail Products with TensorFlow 2. *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, 5(4), 144 - 152. <https://doi.org/10.62527/jitsi.5.4.298>
- Alfikri, R. H., Utomo, M. S., Februariyanti, H., & Nurwahyudi, E. (2022). Pembangunan aplikasi penerjemah bahasa isyarat dengan metode CNN berbasis Android. *Jurnal Teknoinfo*, 16(2), 123-130.
- Colón, R., & Jonathan, D. (2023). A Personal Health Monitoring and Emergency Assistance Mobile Application. *Computer Engineering*;
- Gumelar, G., Hafiar, H., & Subekti, P. (2018). Konstruksi makna Bisindo sebagai budaya tuli bagi anggota Gerkatin. *Jurnal Informasi*, 48(1), 65-78.
- Jagtap, S., Jadhav, K., Temkar, R., & Deshmukh, M. (2024). Real-time Sign Language Recognition Using MobileNetV2 and Transfer Learning. *arXiv preprint arXiv:2412.07486*.
- Jonathan, M., & Rakun, E. (2022). Translating SIBI (Sign System for Indonesian Gesture) Gesture-to-Text in Real-Time using a Mobile Device. *Journal of ICT Research & Applications*, 16(3).
- Lu, J., Liu, X., Ma, X., Tong, J., & Peng, J. (2023). Improved MobileNetV2 crop disease identification model for intelligent agriculture. *PeerJ Computer Science*, 9, e1595. <https://doi.org/10.7717/peerj-cs.1595>
- Luo, C., He, X., Zhan, J., Wang, L., Gao, W., & Dai, J. (2020). Comparison and benchmarking of ai models and frameworks on mobile devices. *arXiv preprint arXiv:2005.05085*.
- Purwono, P., Ma'arif, A., Rahmani, W., Fathurrahman, H., Frisky, A., & Haq, Q. (2023). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2(4), 739-748. doi: <https://doi.org/10.31763/ijrcs.v2i4.888>

- Rajput, A. S., Rajput, D. A. S., Shukla, D. S., & Thakur, D. S. Cutting-Edge Artificial Intelligence in Agriculture: Deepfusionnet Model for Tomato Leaf Disease Classification. *Available at SSRN 4942525*.
- Rao, B. P., Swamy, M. P., & Jain, V. INTEGRATION OF SPEECH RECOGNITION AND MACHINE LEARNING TECHNOLOGIES INTO ONLINE PUBLIC ACCESS CATALOGS FOR LIBRARY SERVICES.
- Ryumin, D., Ivanko, D., & Ryumina, E. (2023). Audio-Visual Speech and Gesture Recognition by Sensors of *Mobile* Devices. *Sensors*, 23(4), 2284. <https://doi.org/10.3390/s23042284>
- Santoso, C., Singadji, M., Purnama, D., Abdel, S., & Kharismawardani, A. (2024). Enhancing Apple Leaf Disease Detection with *Deep Learning*: From Model Training to Android App Integration. *Journal of Applied Data Sciences*, 6(1), 377-390. doi:<https://doi.org/10.47738/jads.v6i1.507>
- Srivastava, A. (2024). *Improving human-robot spoken interactions through enhanced speech-to-text using Whisper* (Master's dissertation). Ghent University – imec.
- Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*, 11(3), 52. <https://doi.org/10.3390/computation11030052>
- Zhou, D., Hou, Q., Chen, Y., Feng, J., & Yan, S. (2020). Rethinking bottleneck structure for efficient *mobile* network design. In A. Vedaldi, H. Bischof, T. Brox, & J. M. Frahm (Eds.), *Computer vision – ECCV 2020. ECCV 2020. Lecture notes in computer science* (Vol. 12348). Springer, Cham.

SK DOSEN PEMBIMBING TUGAS AKHIR



Universitas Sumatera Utara
Fakultas Ilmu Komputer dan
Teknologi Informasi

Alamat
Jalan Universitas
No. 9 Kampus USU,
Medan 20155
Email:
fasilkomti@usu.ac.id
Telepon: (061) 8213793

SURAT KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
NOMOR: 1534/UN5.2.14.D/SK/HK.07/2025

Tentang
Susunan Personalia Dosen Pembimbing Tugas Akhir Mahasiswa Program Studi (S-1) Ilmu Komputer
Fakultas Ilmu Komputer dan Teknologi Informasi (Fasilkom-TI) Universitas Sumatera Utara

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi USU

Menimbang : Bawa Tugas Akhir adalah karya ilmiah tertulis dan/atau prototipe, atau proyek, baik secara individu maupun berkelompok sebagai syarat penyelesaian studi akademik, maka dipandang perlu untuk menetapkan Dosen Pembimbing Penyusunan Tugas Akhir mahasiswa (i) yang bersangkutan.

Mengingat : 1. Undang-undang Nomor 20 tahun 2003 tentang Sistem Pendidikan Nasional;
2. Peraturan Pemerintah Nomor: 48 tahun 1957 tentang Penetapan Pendirian USU;
3. Peraturan Pemerintah Nomor: 19 tahun 2005 tentang Standar Nasional Pendidikan;
4. Peraturan Pemerintah Nomor: 17 tahun 2010 tentang Pengelolaan dan Penyelenggaraan Pendidikan;
5. Keputusan Rektor USU;
a. Nomor: 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana (S1) USU;
c. Nomor: 1876/UN5.1.R/SK/SDM/2021 tentang Pengangkatan Dekan Fasilkom-TI USU Periode 2021-2026.
6. Keputusan Rektor Nomor 459/UN5.1.R/SK/SPB/2023 tentang Pedoman Pelaksanaan Program Merdeka Belajar Universitas Sumatera Utara.
7. Keputusan Dekan Nomor 2968/UN5.2.1.14/SPB/2023 tentang Pedoman Rekognisi Prestasi Akademik dan Non-Akademik Kegiatan Mahasiswa dan Pelaksanaan Program Merdeka Belajar.

Membaca : Hasil persetujuan Dosen Wali Akademik tentang Bentuk Lain Setara Skripsi mahasiswa yang bersangkutan oleh Fakultas Ilmu Komputer dan Teknologi Informasi USU tanggal 20 Maret 2025 dengan judul :
"Integrasi Model Deep Learning dengan Pre-trained MobileNetV2 dan Speech Recognition pada Aplikasi Video Call ElCue untuk Pendekripsi Bahasa Isyarat Indonesia (BISINDO)"

Menetapkan

Menetapkan : Susunan Personalia Pembimbing Tugas Akhir Seorang Mahasiswa Program Studi (S-1) Ilmu Komputer Fasilkom TI USU Medan sebagai berikut:

1. Mahasiswa terbimbing adalah :
Nama : Helga Priscilla Br Purba
NIM : 211401067
Program Studi : S-1 Ilmu Komputer Fasilkom-TI USU
2. Dosen Pembimbing:
1. Hayatunnu'fis, S.Kom., M.Cs. (Pembimbing Pertama)
NIP 199207192024062001
2. Dr. Maya Silvi Lydia, B.Sc., M.Sc. (Pembimbing Kedua)
NIP 197401272002122001
3. Surat Keputusan ini berlaku selama 6 (enam) bulan sejak tanggal **21 Maret 2025 sampai dengan 21 September 2025**. Apabila mahasiswa belum menyelesaikan tugas akhir dalam waktu tersebut, maka Surat Keputusan ini dapat dievaluasi kembali.
4. Segala sesuatu akan diperbaiki kembali, jika di kemudian hari temyata terdapat kekeliruan dalam Surat Keputusan ini.

Medan, 24 Maret 2025
Ditandatangani secara elektronik oleh:
Dekan



Tembusan:
1. Dosen Pembimbing Tugas Akhir
2. Yang bersangkutan

Maya Silvi Lydia
NIP 197401272002122001

SK KEPALA BPTI PUSPRESNAS KEMENDIKBUD RISTEK NOMOR
1653/J7.1/PN.00/2024

No	Kelas	Ket	Nama	Perguruan Tinggi	Judul
23.	PKM-PM4	Perak	KHAIRUNNISA' NUR FADHILAH	Universitas Negeri Surabaya	Metasekerta: Action-Adventure Game Bertema Kesenian Wayang Wong untuk Mengoptimalkan Culture-Based Learning di SMP PGRI 1 Buduran Sidoarjo
24.	PKM-PM4	Perunggu	AULIA NURUL HIKMAH	Institut Pertanian Bogor	Rintik Muda: Penguatkan Self-Control dengan Metode Ant-Fection sebagai Upaya Pengembangan Diri pada Anak Yayasan Titipan Ummat Indonesia
25.	PKM-KC1	Emas	MUHAMMAD ADAM WILDAN	Universitas Airlangga	AUVEST: Automatic Vest with Cooling and Heating System Terintegrasi Aplikasi Mobile Sebagai Upaya Pencegahan Heat Stress Pada Pekerja Konstruksi
26.	PKM-KC1	Perak	BURHANUDIN YUSUF ABDULLAH AR RAMADHAN	Universitas Negeri Malang	Desain dan Implementasi Sistem Teknologi Berbantuan Virtual Reality dan Exoskeleton sebagai Perangkat Rehabilitasi Pasca Stroke
27.	PKM-KC1	Perunggu	SETIYAKI ARUMA NANDI	Universitas Brawijaya	Sistem Rekomendasi Tanaman Pertanian Berbasis Remote Sensing dan Machine Learning dengan Data Citra Satelit Sentinel-2A Berdasarkan Karakteristik Tanah
28.	PKM-KC2	Emas	FRENGKI PRABOWO SAPUTRO WIJAYANTO	Universitas Gadjah Mada	EMO-vest: Futuristic Vest as Epilepsy Detector with Airguard Protecting Based on IoT
29.	PKM-KC2	Perak	KRISNA SEIYA EKIAWAN	Universitas Brawijaya	Alat Deteksi Dini Bladder Cancer Berbasis Quantum Dots dengan Fluorescence Resonance Energy Transfer Terintegrasi Application Programming Interface Klasifikasi Fuzzy Logic
30.	PKM-KC2	Perunggu	MOCH. AVIN	Institut Teknologi Sepuluh Nopember	Integrated Otoscope Examination melalui Klasifikasi Citra Tympanic Membrane untuk Deteksi Otitis Media sebagai Preventif Conductive Hearing Loss berbasis Deep Learning
31.	PKM-KC3	Emas	TESSA AGITHA IRWANI BR BARUS	Universitas Sumatera Utara	ElCue : Aplikasi Video Call dengan Pendekripsi Bahasa Isyarat Indonesia (BISINDO) Berbasis Computer Vision dan Deep Learning untuk Meningkatkan Komunikasi Inklusif
32.	PKM-KC3	Perak	RAISA ZAHRA SALASABILA	Politeknik Negeri Madiun	Sistem Identifikasi dan Monitoring Potensi Bahaya dan Kelalaian Perilaku Pekerja Mesin Gerinda Berbasis Deep Learning Upaya Mencegah Kecelakaan Kerja
33.	PKM-KC3	Perunggu	WAHYU PUTRA	Universitas	Smart Vision (SmarV): Alat Pendekripsi Obat secara Real-Time Berbasis Algoritma Deep

DOKUMENTASI PENGUMUMAN PIMNAS KE-37



SERTIFIKAT PESERTA DAN PEMENANG PIMNAS KE-37



Sertifikat

Nomor 29249/BPTI/DIKTI/2024

Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, Pusat Prestasi Nasional,
Balai Pengembangan Talenta Indonesia menyampaikan penghargaan kepada:

HELGA PRICILLA BR. PURBA

Universitas Sumatera Utara

atas prestasinya sebagai

Peserta bidang **PKM-Karsa Cipta**

pada **Pekan Ilmiah Mahasiswa Nasional (PIMNAS)** ke 37 Tahun 2024 yang dilaksanakan oleh
Balai Pengembangan Talenta Indonesia sebagai unit pelaksana teknis Pusat Prestasi Nasional, Kementerian Pendidikan,
Kebudayaan, Riset, dan Teknologi secara luring bekerja sama dengan Universitas Airlangga,
Provinsi Jawa Timur pada tanggal 14 s.d. 19 Oktober 2024.

Jakarta, 18 Oktober 2024
Plt. Kepala Balai Pengembangan Talenta Indonesia,



Dr. Maria Veronica Irene Herdjono, S.E., M.Si.
NIP 198103292012122001



Sertifikat

Nomor 29252/PPN/DIKTI/2024

Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, Pusat Prestasi Nasional,
Balai Pengembangan Talenta Indonesia menyampaikan penghargaan kepada:

HELGA PRICILLA BR. PURBA

Universitas Sumatera Utara

atas prestasinya sebagai

Medali Emas kategori **Poster**
bidang **PKM-Karsa Cipta**

pada **Pekan Ilmiah Mahasiswa Nasional (PIMNAS)** ke 37 Tahun 2024 yang dilaksanakan oleh
Balai Pengembangan Talenta Indonesia sebagai unit pelaksana teknis Pusat Prestasi Nasional, Kementerian Pendidikan,
Kebudayaan, Riset, dan Teknologi secara luring bekerja sama dengan Universitas Airlangga,
Provinsi Jawa Timur pada tanggal 14 s.d. 19 Oktober 2024.

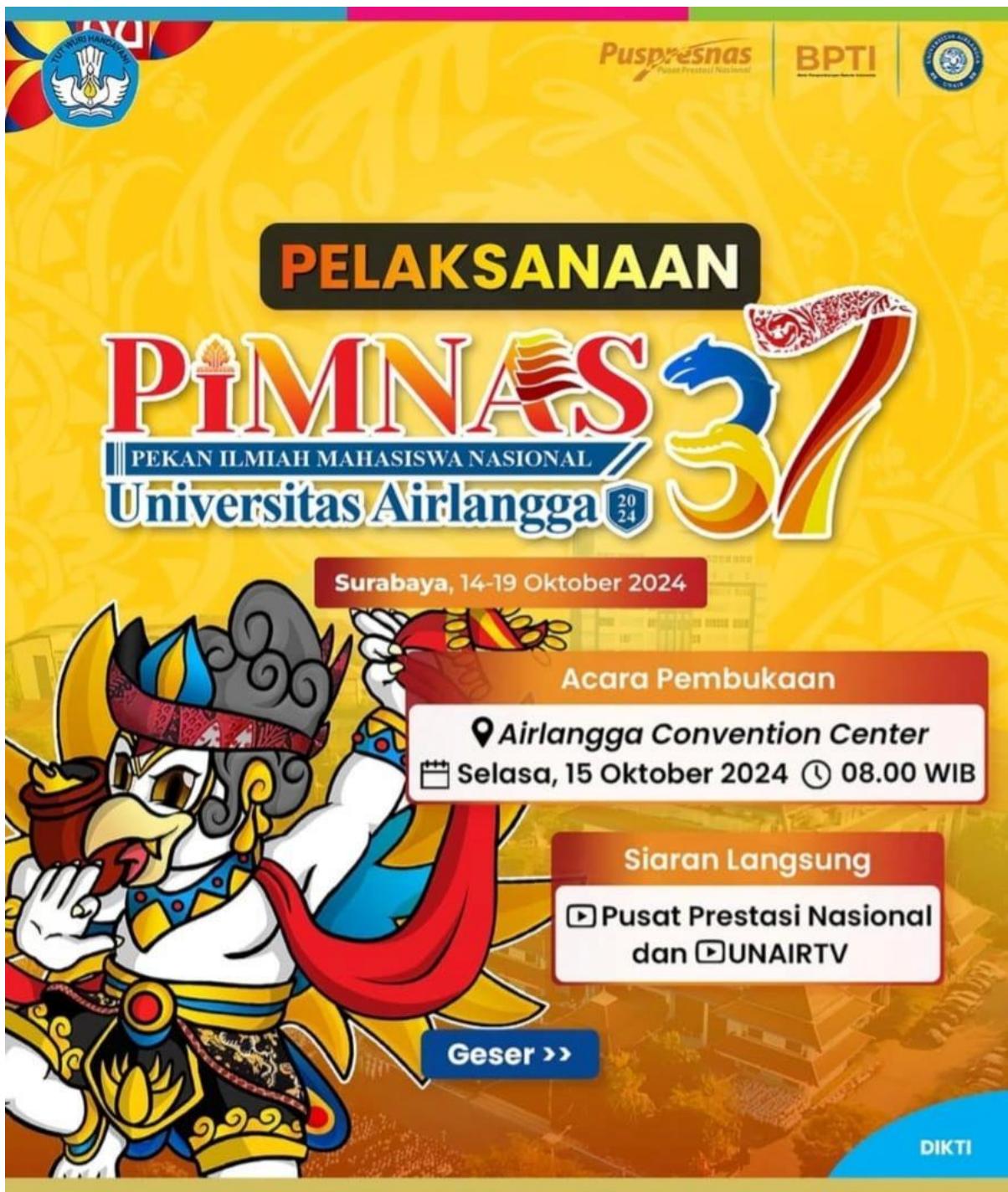
Jakarta, 18 Oktober 2024
Kepala Pusat Prestasi Nasional,



Dr. Maria Veronica Irene Herdjono, S.E., M.Si.
NIP 198103292012122001



FLYER LOMBA PIMNAS KE-37



PENJELASAN PERANAN MAHASISWA DALAM TIM

No	Nama / NIM	Program Studi	Uraian Tugas
1	Tessa Agitha Irwani Br. Barus / 211401138	S-1 Ilmu Komputer	<ul style="list-style-type: none"> Bertugas melakukan <i>fine-tuning</i> pada model MobileNetV2. Mendesain dan menguji tambahan layer (<i>custom layers</i>) untuk meningkatkan akurasi deteksi. Mengukur performa model (akurasi, latency) pada input <i>real-time</i>.
2	Putri Andriyani / 211401008	S-1 Ilmu Komputer	<ul style="list-style-type: none"> Membangun aplikasi ElCue dengan Flutter. Mengintegrasikan layanan backend (Firebase) dan <i>video call</i> (Agora). Menyambungkan model deteksi BISINDO ke tampilan aplikasi secara <i>real-time</i>.
3	Rani Widya Astuti / 211401018	S-1 Ilmu Komputer	<ul style="list-style-type: none"> Meneliti dan menerapkan teknik <i>pre-processing</i> berupa augmentasi data dan <i>edge detection</i>. Memastikan model tetap robust terhadap variasi gerakan tangan dalam kondisi <i>real-time</i>. Bekerja sama membuat model untuk memastikan hasil <i>pre-processing</i> optimal.
4	Angela Siadari / 211401030	S-1 Ilmu Komputer	<ul style="list-style-type: none"> Merancang dan mengembangkan sistem pengumpulan dataset BISINDO (video & gambar). Bertanggung jawab atas <i>labeling</i>, anotasi data, dan validasi kualitas data. Mengembangkan pipeline inferensi berbasis <i>Computer Vision</i> untuk integrasi ke sistem ElCue.
5	Helga Pricilla Br. Purba / 211401067	S-1 Ilmu Komputer	<ul style="list-style-type: none"> Mengintegrasikan hasil dari model <i>deep learning</i> dan <i>speech recognition</i> ke sistem ElCue. Mengatur alur komunikasi antar input <i>gesture</i> dan suara untuk meningkatkan komunikasi 2 arah. Menjamin sinkronisasi dan performa dari dua jalur input (<i>gesture + voice</i>).

HASIL TURNITIN

Bentuk_lain_setara_skripsi-1744076103841

ORIGINALITY REPORT

17%
SIMILARITY INDEX **16%**
INTERNET SOURCES **8%**
PUBLICATIONS **11%**
STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Sumatera Utara Student Paper	2%
2	repositori.usu.ac.id Internet Source	2%
3	repositori.usu.ac.id:8080 Internet Source	2%
4	123dok.com Internet Source	1%
5	text-id.123dok.com Internet Source	1%
6	Indah Clara Sari. "INTEGRASI MODEL DEEP LEARNING EFFICIENTNET-B0 UNTUK DETEKSI PENYAKIT DAUN TOMAT PADA APLIKASI SELULER BERBASIS FLUTTER", Djtechno: Jurnal Teknologi Informasi, 2024 Publication	<1 %
7	www.coursehero.com Internet Source	<1 %
8	link.springer.com Internet Source	<1 %
9	Submitted to University of Stirling Student Paper	<1 %
10	id.123dok.com Internet Source	<1 %