

**OPTIMASI MOBILENETV2 DENGAN *FINE-TUNING* DAN
ADDITIONAL LAYERS UNTUK Pendeteksian Bahasa
ISYARAT INDONESIA (BISINDO) SECARA *REAL-TIME*
DALAM APLIKASI *VIDEO CALL ELCUE***

TUGAS AKHIR DALAM BENTUK LAIN SETARA SKRIPSI

**TESSA AGITHA IRWANI BR BARUS
211401138**



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

**OPTIMASI MOBILENETV2 DENGAN *FINE-TUNING* DAN
ADDITIONAL LAYERS UNTUK Pendeteksian Bahasa
ISYARAT INDONESIA (BISINDO) SECARA *REAL-TIME*
DALAM APLIKASI *VIDEO CALL ELCUE***

TUGAS AKHIR DALAM BENTUK LAIN SETARA SKRIPSI

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

TESSA AGITHA IRWANI BR BARUS

211401138



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

PERSETUJUAN

Judul : OPTIMASI MOBILENETV2 DENGAN *FINE-TUNING* DAN *ADDITIONAL LAYERS* UNTUK PENDETEKSIAN BAHASA ISYARAT INDONESIA (BISINDO) SECARA *REAL-TIME* DALAM APLIKASI *VIDEO CALL ELCUE*

Kategori : TUGAS AKHIR DALAM BENTUK LAIN SETARA SKRIPSI

Nama : TESSA AGITHA IRWANI BR BARUS

Nomor Induk Mahasiswa : 211401138

Program Studi : S-1 ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Komisi Pembimbing :

Medan, 27 Maret 2025

Pembimbing II

Dr. Mohammad Andri Budiman
S.T., M.Comp.Sc., M.E.M.
NIP. 197510082008011011

Pembimbing I

Hayatunnufus, S.Kom., M.Cs
NIP. 199207192024062001

Diketahui/disetujui oleh
Program Studi S-1 Ilmu Komputer



Dr. Amalia S.T., M.T.

NIP. 197812212014042001

PERNYATAAN

OPTIMASI MOBILENETV2 DENGAN *FINE-TUNING* DAN *ADDITIONAL LAYERS* UNTUK PENDETEKSIAN BAHASA ISYARAT INDONESIA
(BISINDO) SECARA *REAL-TIME* DALAM
APLIKASI VIDEO CALL ELCUE

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, 2025


Tessa Agitha Irwani Br Barus
211401138

PENGHARGAAN

Puji dan Syukur penulis ucapkan ke hadirat Tuhan Yang Maha Esa yang senantiasa memberikan berkat, kasih, serta penyertaan-Nya sehingga penulis dapat menyelesaikan bentuk lain setara skripsi sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada program studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

Penulis juga mengucapkan terima kasih sebesar-besarnya kepada semua pihak atas segala dukungan, bantuan, doa yang diberikan kepada penulis dalam menyelesaikan bentuk lain setara skripsi ini. Adapun pada kesempatan kali ini penulis hadiahkan kepada kedua orang tua penulis, Bapak Nirwan Barus dan Ibu Almina Ginting yang tidak pernah memberatkan penulis perihal tugas akhir serta selalu melimpahkan kasih sayang, waktu, dan materi yang tak terhingga. Selain itu, penulis sampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yesus Kristus yang senantiasa memberikan kasih dan berkat-Nya kepada penulis sehingga dapat menyelesaikan bentuk lain setara skripsi ini dengan baik.
2. Bapak Dr. Muryanto Amin, S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
3. Ibu Dr. Maya Silvi Lidya, M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
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, Dosen Pembimbing II, dan Dosen Pembimbing Akademik yang telah memberi banyak dukungan, motivasi, masukan, dan bimbingan yang membangun kepada penulis selama penyusunan skripsi ini.
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 yang dengan penuh kesabaran, dukungan, motivasi, serta masukan konstruktif yang diberikan, telah membimbing dan memberikan arahan yang sangat berarti dalam pengembangan proyek ElCue dan

penyusunan bentuk lain setara skripsi ini, serta atas bimbingan spiritual yang telah membangun semangat dan keteguhan hati penulis untuk terus berusaha memberikan yang terbaik.

7. Bapak dan Ibu Dosen Program Studi S1 Ilmu Komputer Universitas Sumatera Utara yang telah membagi ilmu, wawasan, dan pengalaman kepada penulis selama masa perkuliahan.
8. Staf dan pegawai Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang siap sedia membantu urusan administrasi perkuliahan hingga dalam menyelesaikan bentuk lain setara skripsi ini.
9. Kedua Orang Tua penulis, Bapak, Nirwan Barus S.Pd. dan Mamak, Dra. Almina Ginting yang selalu memberikan dukungan penuh, membantu, mendo'akan, materi yang tak terhingga, serta tidak pernah memberatkan penulis perihal penyelesaian bentuk lain setara skripsi ini.
10. Kakak penulis, Yessikha Valerine Irwani, dan Abang, Jeffery Irwana dan Rio Crisvan Irwana yang selalu mendukung penuh segala kegiatan penulis dan memberikan semangat.
11. Sahabat-sahabat sekaligus partner perjuangan dalam berbagai kegiatan akademik maupun non akademik di dalam maupun di luar kampus, dari keluarga Potobut, Helga Pricilla Br. Purba, Angela Siadari, Rani Widya Astuti, Putri Andriyani dan Imanda Tamara Br. Pasaribu, yang dengan tulus selalu memberikan ilmu-ilmu positif, masukan yang berharga, serta bantuan dan kemudahan yang sangat berarti selama perjalanan perkuliahan, dan kehadiran mereka sebagai teman sejati yang senantiasa memberi semangat dan motivasi telah memperkaya pengalaman penulis.
12. Sahabat penulis dari masa SMA, Mesiya Anastasia Barus yang senantiasa mendukung serta mendo'akan segala kegiatan penulis.
13. Teman-teman dari Program Kreativitas Mahasiswa (PKM) hingga PIMNAS 37 kontingen Universitas Sumatera Utara yang telah memberikan dukungan, semangat, dan kerja sama yang luar biasa sepanjang perjalanan kegiatan PKM dan PIMNAS.
14. Teman-teman Kom B dan angkatan 2021, Haikal, Lorenzo, Harry, Samuel, Pieter, Elisa, Muthia, Reysha, yang telah memberikan peran selama masa

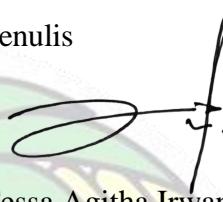
perkuliahan di program studi Ilmu Komputer.

15. Seluruh kerabat dan teman-teman lainnya yang tidak bisa disebutkan satu persatu yang telah memberikan dukungan dan banyak saran dalam hal apapun.

Semoga bantuan, dukungan serta doa-doa yang dipanjatkan dibalas dengan kebaikan yang lebih baik oleh Tuhan Yang Maha Esa.

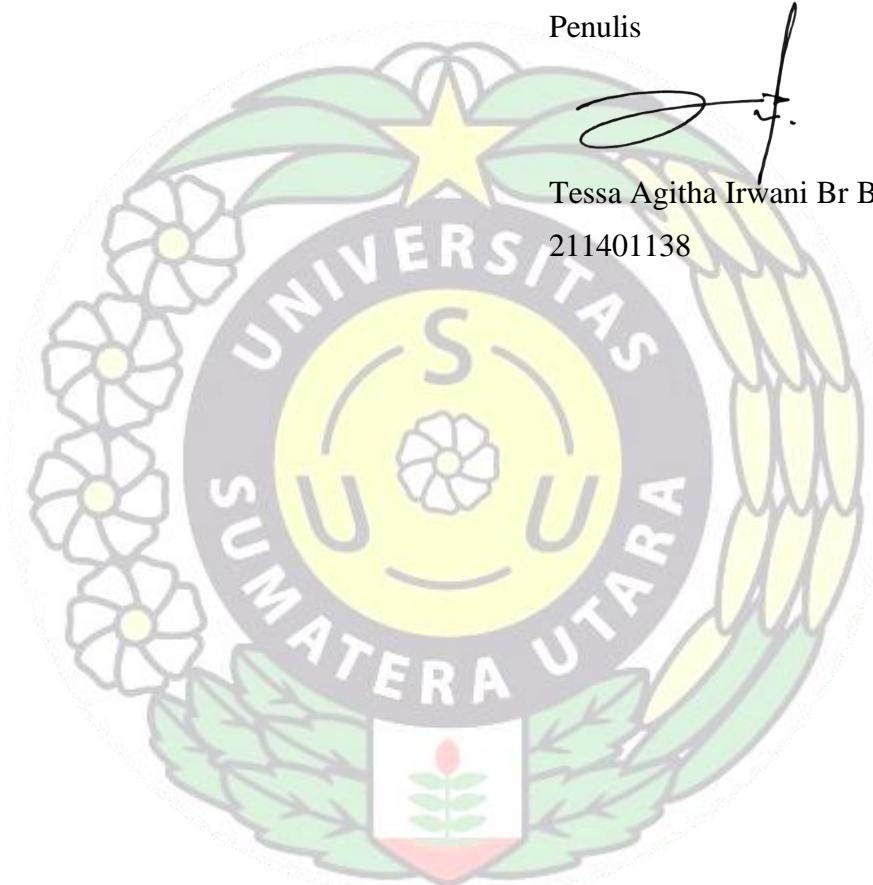
Medan, 26 Maret 2025

Penulis



Tessa Agitha Irwani Br Barus

211401138



ABSTRAK

Kemajuan teknologi komunikasi seperti *video call* telah mempermudah interaksi jarak jauh. Namun, penyandang disabilitas Tuli di Indonesia masih mengalami hambatan komunikasi karena kurangnya pemahaman terhadap Bahasa Isyarat Indonesia (BISINDO). Penelitian ini mengembangkan sistem pendekripsi BISINDO secara *real-time* dalam aplikasi *video call* ElCue dengan mengoptimalkan MobileNetV2 menggunakan teknik *fine-tuning* dan penambahan layer. Metode penelitian mencakup pengumpulan dataset BISINDO dengan 9 kelas gestur, *preprocessing* melalui augmentasi, *edge detection*, dan normalisasi, serta pelatihan model menggunakan *transfer learning* dan *fine-tuning* dengan membuka beberapa layer terakhir MobileNetV2 untuk menyesuaikan bobot. Model dikembangkan dengan tambahan layer Global Average Pooling, Dense Layer, dan Dropout untuk meningkatkan akurasi dan stabilitas klasifikasi. Pengujian dilakukan terhadap akurasi model serta performa inferensi *real-time* menggunakan kamera. Hasil penelitian menunjukkan model mencapai akurasi 97.1% pada data uji dan 88.9% dalam pengujian *real-time*, yang membuktikan bahwa MobileNetV2 yang telah dioptimasi dapat menjadi solusi efektif dalam mendekripsi BISINDO secara *real-time*.

Kata Kunci: MobileNetV2, BISINDO, *Fine-tuning*, *Deep Learning*, *Real-time Detection*

**OPTIMIZATION OF MOBILENETV2 THROUGH FINE-TUNING AND
ADDITIONAL LAYERS FOR REAL-TIME DETECTION OF INDONESIAN
SIGN LANGUAGE (BISINDO) IN THE ELCUE VIDEO CALL
APPLICATION**

ABSTRACT

Advancements in communication technology, such as video calls, have facilitated long-distance interactions. However, individuals with hearing disabilities still face communication barriers due to the limited understanding of Indonesian Sign Language (BISINDO) among public. This study proposes a real-time BISINDO detection system in the ElCue video call application by optimizing MobileNetV2 through fine-tuning techniques and the incorporation of additional layers. The research methodology involves collecting BISINDO dataset consisting of 9 gesture classes, preprocessing through augmentation, edge detection, and normalization, followed by model training utilizing transfer learning and fine-tuning, wherein several final layers of MobileNetV2 are unfrozen to refine the model's weights. Additionally, the model architecture is extended with Global Average Pooling, Dense Layers, and Dropout to enhance classification accuracy and stability. The model's performance is evaluated based on accuracy and real-time inference capabilities using a camera. The results indicate that the optimized model achieved 97.1% accuracy on the test dataset and 88.9% in real-time testing, proving that the optimized MobileNetV2 can serve as an effective solution for real-time BISINDO detection.

Keywords: MobileNetV2, BISINDO, Fine-tuning, Deep Learning, Real-time Detection

DAFTAR ISI

PERSETUJUAN	i
PERNYATAAN.....	ii
PENGHARGAAN	iii
ABSTRAK.....	vi
<i>ABSTRACT</i>	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	xi
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	5
1.6.5 Pengujian Sistem	5
1.6.6 Penyusunan Laporan	6
1.7 Sistematika Penulisan	6
BAB 2 LANDASAN TEORI.....	8
2.1 Bahasa Isyarat Indonesia (BISINDO).....	8
2.2 <i>Deep Learning</i>	8
2.3 <i>Convolutional Neural Network</i> (CNN)	9
2.3.1 Lapisan Konvolusi (Convolutional Layer).....	9
2.3.2 Pooling Layer	10
2.3.3 Fungsi Aktivasi (Activation Function)	11
2.3.4 Fully Connected Layer	12
2.4 <i>Pre-trained MobileNet</i>	13
2.5 <i>Fine-tuning</i> untuk Optimasi Model CNN	15
2.5.1 Additional Layers	16

2.5.2 Regularization	16
2.5.3 Loss Function	17
2.5.3 Learning Rate	18
2.5.4 Optimizer.....	19
2.6 Evaluasi Model.....	20
2.6.1 Metrik Evaluasi Model.....	20
2.6.2 Confusion Matrix	21
2.7 Penelitian Terkait	22
BAB 3 ANALISIS DAN PERANCANGAN	25
3.1 Analisis Sistem.....	25
3.1.1 Analisis Masalah	25
3.1.2 Analisis Data	26
3.1.3 Analisis Kebutuhan	27
3.1.4 Diagram Umum Sistem.....	29
3.2 Pemodelan Sistem	32
3.2.1 Activity Diagram.....	32
3.2.2 Flowchart	33
BAB 4 IMPLEMENTASI DAN PENGUJIAN	36
4.1 Implementasi	36
4.1.1 Implementasi Model Deep Learning.....	36
4.1.2 Implementasi Model secara Real-time.....	41
4.2 Pengujian.....	43
4.2.1 Pengujian Akurasi Model.....	43
4.2.2 Pengujian Performa secara Real-time	47
BAB 5 KESIMPULAN DAN SARAN	49
5.1 Kesimpulan	49
5.2 Saran.....	49
DAFTAR PUSTAKA	51
LAMPIRAN	54

DAFTAR TABEL

Tabel 3.1 Spesifikasi Perangkat Lunak	28
Tabel 3.2 Spesifikasi Perangkat Keras	28
Tabel 4.1 Perbandingan Hasil Evaluasi Model dengan Penelitian Sebelumnya..	46
Tabel 4.2 Performa Model ketika Inferensi Real-time	47



DAFTAR GAMBAR

Gambar 2.1 Isyarat BISINDO (Sumber: Yayasan Peduli Kasih ABK, 2019)	8
Gambar 2.2 Arsitektur CNN secara Sederhana (Sumber: Phung & Rhee, 2019) .	9
Gambar 2.3 Proses Operasi pada Convolutional Layer (Sumber: IBM, 2023)...	10
Gambar 2.4 Operasi Max Pooling (Sumber: Peryanto, dkk., 2020).....	11
Gambar 2.5 Fully Connected Layer (Sumber: Kalayci & Asan, 2022)	13
Gambar 2.6 Arsitektur MobileNet (Sumber: Howard dkk., 2017).....	14
Gambar 2.7 Arsitektur MobileNetV2 (Sumber: Sandler dkk., 2019)	15
Gambar 3.1 Diagram Ishikawa.....	26
Gambar 3.2 Dataset BISINDO	27
Gambar 3.3 Diagram Umum Sistem	29
Gambar 3.4 Arsitektur Model Deteksi BISINDO ElCue dengan MobileNetV2 dan Additional Layers	30
Gambar 3.5 Arsitektur Model Deteksi BISINDO ElCue dengan MobileNetV2 yang telah dioptimasi dengan fine-tuning dan additional layers.....	31
Gambar 3.6 Activity Diagram	33
Gambar 3.7 Flowchart	35
Gambar 4.1 Code untuk Memuat Dataset BISINDO dari Kaggle	36
Gambar 4.2 Code untuk Tahap Preprocessing Data.....	38
Gambar 4.3 Code untuk Membangun Fine-tuning Model dengan MobileNetV2 dan Additional Layers	40
Gambar 4.4 Code untuk Training Model.....	41
Gambar 4.5 Code untuk konversi model ke TFLite	41
Gambar 4.6 Code untuk implementasi model secara real-time.....	42
Gambar 4.7 Performa selama proses training dengan 50 epoch.....	44
Gambar 4.8 Grafik training and validation untuk loss dan accuracy	44
Gambar 4.9 Evaluasi Model	45
Gambar 4.10 Confusion Matrix.....	46
Gambar 4.11 Confusion Matrix.....	46
Gambar 4.12 Inferensi Model secara Real-time	48

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Kemajuan pesat dalam teknologi informasi dan komunikasi telah membawa perubahan besar dalam berbagai aspek kehidupan, termasuk cara individu berkomunikasi. Salah satu inovasi terbesar dalam bidang telekomunikasi adalah kemunculan *video call* yang memungkinkan interaksi langsung meskipun berada di lokasi yang berbeda. *Video call* mempermudah komunikasi dengan memungkinkan pertukaran informasi secara *real-time* dengan cara yang lebih alami dan interaktif dibandingkan bentuk komunikasi teks atau suara saja.

Walaupun teknologi ini telah berkembang luas, tidak semua kelompok masyarakat dapat memanfaatkannya secara maksimal. Penyandang disabilitas, terutama penyandang disabilitas Tuli, masih mengalami hambatan dalam berkomunikasi dengan kelompok non-Tuli, meskipun kemajuan teknologi komunikasi terus berlanjut. Berdasarkan data statistik yang diterbitkan oleh BPS pada tahun 2022, terdapat sekitar 2,1 juta penyandang disabilitas Tuli di Indonesia. Data statistik sekolah luar biasa pada tahun 2019 mencatat hanya 26.438 penyandang Tuli. Hal ini memaikan peran krusial dalam memahami satu sama lain dalam kehidupan bermasyarakat. Kendati demikian, aksesibilitas komunikasi bagi kelompok ini masih terbatas, terutama dalam hal penggunaan bahasa isyarat dalam interaksi sosial sehari-hari. Bahasa Isyarat Indonesia (BISINDO), yang menjadi sarana komunikasi utama bagi penyandang disabilitas Tuli di Indonesia, masih belum dipahami secara luas oleh masyarakat non-Tuli (Purnamasari, 2021).

BISINDO adalah bahasa yang dipergunakan oleh masyarakat Tuli di Indonesia untuk berkomunikasi. Meskipun telah diakui secara resmi oleh pemerintah pada tahun 1996, BISINDO masih kurang dipahami oleh banyak kalangan non-Tuli. Hal ini menyebabkan hambatan dalam komunikasi antara kelompok Tuli dan non-Tuli, yang berpotensi menimbulkan kesalahpahaman dan isolasi sosial bagi penyandang disabilitas Tuli (Rizky et al., 2023). Kurangnya akses terhadap pelatihan bahasa isyarat di masyarakat umum dan ketidaktersediaan alat

bantu komunikasi yang efektif menjadi tantangan besar dalam mewujudkan inklusivitas komunikasi di masyarakat.

Dalam konteks ini, aplikasi ElCue hadir sebagai solusi inovatif yang menggabungkan teknologi *video call* dengan kemampuan deteksi bahasa isyarat secara *real-time* menggunakan teknologi *computer vision* dan *deep learning*. Teknologi ini memungkinkan aplikasi untuk mendeteksi dan menginterpretasikan gerakan BISINDO melalui kamera seluler, yang kemudian diterjemahkan menjadi teks atau suara yang dapat dimengerti oleh individu non-Tuli. Teknologi *deep learning* memungkinkan aplikasi untuk mempelajari variasi gerakan BISINDO secara otomatis, meningkatkan akurasi dan fleksibilitas sistem dalam beradaptasi dengan penggunaan bahasa isyarat yang beragam.

Berbagai penelitian telah dilakukan untuk mengembangkan sistem pengenalan bahasa isyarat menggunakan *deep learning*, termasuk yang menggunakan MobileNetV2 sebagai model utama. Salah satu penelitian yang dilakukan oleh Pu & Yi (2024) membandingkan performa EfficientNetB0, MobileNetV2, dan MobileNetV3 Small dalam mendeteksi huruf alfabet *American Sign Language* (ASL) dengan dataset terbatas. Hasil penelitian menunjukkan bahwa MobileNetV2 unggul dalam tahap awal pelatihan, tetapi mengalami fluktuasi performa di *epoch* yang lebih tinggi. Sementara itu, EfficientNetB0 dan MobileNetV3 mengalami *overfitting*, dengan akurasi pelatihan yang tinggi tetapi hasil pengujian yang rendah.

Selain itu, penelitian lain menunjukkan bahwa MobileNetV2 juga telah berhasil diterapkan dalam sistem pengenalan bahasa isyarat secara *real-time*. Studi yang meneliti *Indian Sign Language* (ISL) mengusulkan pendekatan berbasis MobileNetV2 dengan transfer learning, yang dirancang agar ringan dan efisien untuk perangkat *mobile* (Tamkar dkk., 2024). Model ini mencapai akurasi 97% dengan latensi 60ms per frame, membuktikan efektivitasnya dalam deteksi bahasa isyarat secara *real-time*. Berdasarkan penelitian-penelitian tersebut, MobileNetV2 terbukti memiliki potensi dalam mendeteksi bahasa isyarat, tetapi masih memerlukan optimasi lebih lanjut. Oleh karena itu, penelitian ini akan mengembangkan lebih lanjut pendekatan serupa dengan *fine-tuning* dan

penambahan layer tambahan, agar lebih optimal dalam mendeteksi Bahasa Isyarat Indonesia (BISINDO) secara *real-time* dalam aplikasi *video call*.

Optimasi model MobileNetV2 dengan *fine-tuning* dan penambahan *layer* tambahan menjadi pendekatan yang menjanjikan untuk meningkatkan performa sistem dalam mendeteksi gerakan BISINDO secara *real-time*. MobileNetV2, yang dikenal dengan keunggulannya dalam pengolahan citra dengan efisiensi komputasi yang tinggi, memungkinkan implementasi sistem deteksi bahasa isyarat yang cepat dan akurat meskipun menggunakan perangkat dengan sumber daya terbatas, seperti *smartphone*. Dengan optimasi lebih lanjut, diharapkan aplikasi ElCue dapat memberikan kemudahan akses komunikasi bagi penyandang disabilitas Tuli, serta meningkatkan inklusivitas sosial di Indonesia.

Seiring dengan perkembangan teknologi dan inovasi dalam bidang *computer vision* dan *deep learning*, diharapkan aplikasi ElCue dapat menjadi jembatan yang menghubungkan penyandang disabilitas Tuli dengan masyarakat luas, memfasilitasi komunikasi yang lebih efektif dan inklusif bagi semua pihak.

1.2 Rumusan Masalah

Teknologi komunikasi seperti *video call*, yang mempermudah interaksi jarak jauh, belum sepenuhnya dapat mengakomodasi kebutuhan komunikasi penyandang Tuli. Dalam konteks ini, teknologi pendekripsi objek berbasis *deep learning*, seperti MobileNetV2, berpotensi untuk meningkatkan inklusivitas komunikasi melalui aplikasi seperti ElCue yang memungkinkan deteksi gerakan BISINDO secara *real-time*. Oleh karena itu diperlukan penelitian untuk mengoptimalkan model MobileNetV2 dalam pendekripsi BISINDO secara *real-time* melalui penerapan teknik *fine-tuning* dan *additional layers*, guna meningkatkan akurasi dan efisiensi sistem dalam mendekripsi gerakan-gerakan BISINDO yang beragam.

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengoptimalkan model MobileNetV2 dalam pendekripsi Bahasa Isyarat Indonesia (BISINDO) secara *real-time* pada aplikasi *video call* ElCue, dengan penerapan *fine-tuning* dan penambahan *additional layers* untuk meningkatkan akurasi, efisiensi, dan responsivitas sistem.

1.4 Batasan Penelitian

Untuk menjaga agar penelitian tetap terfokus dan tidak meluas, diperlukan penetapan batasan masalah yang jelas, sebagai berikut:

1. Penelitian ini hanya berfokus pada penggunaan model MobileNetV2 yang dioptimalkan dengan *fine-tuning* dan penambahan *additional layers* untuk pendekripsi Bahasa Isyarat Indonesia (BISINDO) dalam aplikasi *video call* ElCue.
2. Sistem yang dikembangkan hanya akan diuji pada dataset gerakan BISINDO yang terbatas, tanpa memperhitungkan variasi yang lebih luas dari gerakan-gerakan BISINDO dalam kehidupan sehari-hari.
3. Evaluasi kinerja model terbatas pada metrik *accuracy, precision, recall, dan F1-Score* dalam mendekripsi gerakan BISINDO, tanpa mencakup analisis terhadap faktor-faktor eksternal seperti perangkat keras yang digunakan maupun aspek latensi sistem dalam pemrosesan *real-time*.
4. Penelitian ini tidak mencakup implementasi sistem dalam skenario penggunaan dunia nyata secara luas, melainkan hanya berfokus pada pengembangan dan evaluasi teknis dalam lingkungan uji coba yang telah ditentukan.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini, yaitu:

1. Membantu mengoptimalkan pendekripsi Bahasa Isyarat Indonesia (BISINDO) secara *real-time* menggunakan model MobileNetV2 yang telah disesuaikan (*fine-tuning*) dan ditambahkan *layer*, sehingga meningkatkan akurasi dan responsivitas sistem dalam aplikasi video call.
2. Diharapkan meningkatkan inklusivitas komunikasi antara penyandang disabilitas Tuli dan non-Tuli di Indonesia melalui aplikasi *video call* ElCue, dengan memungkinkan deteksi otomatis gerakan BISINDO secara *real-time*.
3. Mendukung terciptanya interaksi sosial yang lebih inklusif bagi penyandang disabilitas Tuli, mengurangi hambatan komunikasi, serta memberikan solusi bagi kebutuhan komunikasi mereka dalam berbagai aspek kehidupan sehari-hari.

1.6 Metodologi Penelitian

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

1.6.1 Studi Literatur

Tahap pertama adalah melakukan studi literatur untuk mengumpulkan informasi yang relevan mengenai teori-teori dasar, metode, serta penelitian terdahulu terkait dengan pengenalan bahasa isyarat menggunakan teknologi deep learning. Studi literatur ini mencakup MobileNetV2, transfer learning, serta penerapan model *Convolutional Neural Network* (CNN) dalam deteksi bahasa isyarat.

1.6.2 Analisis Permasalahan

Tahap ini melibatkan analisis masalah yang terkait dengan pengembangan sistem deteksi bahasa isyarat secara real-time dengan akurasi tinggi serta pengoptimalan model deep learning untuk memastikan kinerja yang baik pada perangkat dengan sumber daya terbatas.

1.6.3 Perancangan Sistem

Setelah menganalisis permasalahan, selanjutnya dilakukan proses perancangan sistem yang meliputi pemilihan arsitektur model dan strategi *training* untuk mencapai akurasi yang tinggi sambil mengurangi *overfitting* pada dataset terbatas.

1.6.4 Implementasi

Pada tahap implementasi, dilakukan pengembangan model deteksi bahasa isyarat Indonesia menggunakan MobileNetV2. Model MobileNetV2 akan dioptimalkan melalui fine-tuning serta penambahan lapisan tambahan untuk meningkatkan akurasi dalam mendeteksi berbagai gerakan tangan dalam BISINDO.

1.6.5 Pengujian Sistem

Tahap pengujian sistem dilakukan untuk mengevaluasi keakuratan deteksi BISINDO dalam aplikasi *video call* ElCue. Pengujian mencakup beberapa skenario, yaitu akurasi pengenalan gerakan tangan menggunakan dataset BISINDO untuk mengukur seberapa baik model mengenali gerakan tangan,

kecepatan deteksi *real-time* dengan mengukur waktu respons antara gerakan tangan dan tampilan teks dalam aplikasi.

1.6.6 Penyusunan Laporan

Setelah seluruh tahapan penelitian dan pengujian selesai, tahap penyusunan laporan dilakukan untuk mendokumentasikan proses penelitian, hasil pengujian, serta analisis dan kesimpulan dari penelitian ini.

1.7 Sistematika Penulisan

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

Bab 1: Pendahuluan

Pada bab ini dibahas mengenai latar belakang masalah, rumusan masalah, tujuan, batasan penelitian, manfaat penelitian, dan metode penelitian yang digunakan serta sistematikan penulisan pada penelitian.

Bab 2: Landasan Teori

Bab ini menguraikan teori-teori dasar yang mendasari penelitian, seperti MobileNetV2, *fine-tuning*, dan penambahan *layer* tambahan dalam *deep learning*, serta penerapannya pada deteksi BISINDO. Selain itu, dijelaskan tentang konsep model *deep learning* untuk pengenalan gerakan tangan, serta penelitian terkait aplikasi bahasa isyarat.

Bab 3: Analisis dan Perancangan

Pada bab ini dijelaskan tentang analisis permasalahan yang dihadapi dalam deteksi BISINDO secara *real-time*, termasuk tantangan terkait akurasi, latensi, dan sumber daya terbatas pada perangkat *mobile*. Selain itu, dibahas tentang perancangan sistem, yang meliputi pemilihan model MobileNetV2, *fine-tuning*, dan penambahan *layer* untuk meningkatkan performa sistem dalam aplikasi video call ElCue.

Bab 4: Implementasi dan Pengujian Sistem

Bab ini menguraikan proses training dan pengujian sistem dari penelitian yang sudah dilakukan, agar dapat menunjukkan apakah penelitian ini sudah sesuai dengan hasil dan tujuan yang direncanakan.

Bab 5: Kesimpulan dan Saran

Bab terakhir ini berisi kesimpulan terkait hasil dari penelitian yang telah dilakukan, khususnya mengenai efektivitas deteksi BISINDO menggunakan MobileNetV2

dalam aplikasi *video call*. Selain itu, bab ini juga memberikan saran untuk pengembangan lebih lanjut, baik terkait model *deep learning* maupun perangkat keras yang digunakan, agar aplikasi dapat lebih optimal.



BAB 2

LANDASAN TEORI

2.1 Bahasa Isyarat Indonesia (BISINDO)

Bahasa Isyarat Indonesia (BISINDO) bentuk komunikasi yang secara alami berkembang di komunitas Tuli. Dalam penggunaannya, isyarat tangan dalam BISINDO mencakup abjad, angka, dan kata-kata yang secara langsung mengacu pada konsep atau objek tertentu (Hikmatia dan Zul, 2021). BISINDO sebagai ekspresi linguistik komunitas Tuli, menjadi fokus utama dalam meningkatkan inklusivitas komunikasi melalui aplikasi *video call*. ElCue adalah aplikasi platform *mobile* yang menawarkan fitur *video call* dengan pendekripsi BISINDO, bertujuan untuk meningkatkan komunikasi inklusif antara Tuli dan non-Tuli. Fitur deteksi bahasa isyarat pada ElCue menggunakan *computer vision* dan *deep learning* untuk rekognisi video.



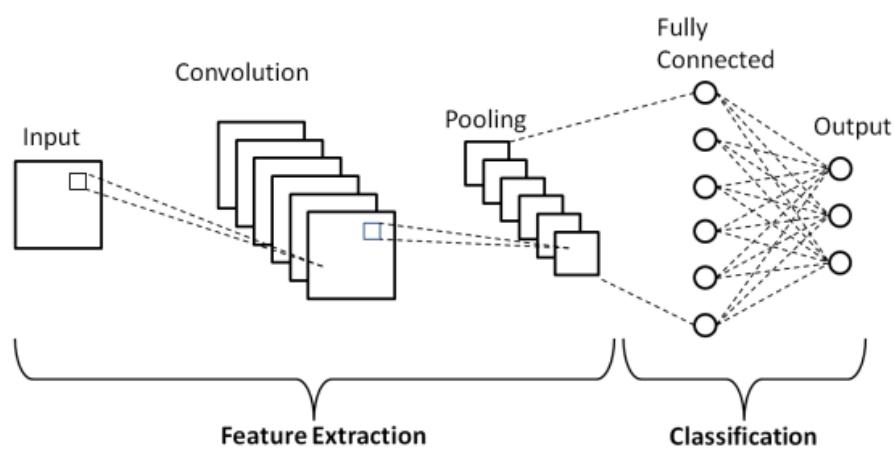
Gambar 2.1 Isyarat BISINDO (Sumber: Yayasan Peduli Kasih ABK, 2019)

2.2 Deep Learning

Deep learning Deep learning adalah cabang dari machine learning yang menggunakan deep neural networks untuk memproses dan menganalisis data. Di dalam deep learning, model jaringan saraf yang rumit digunakan untuk mengekstraksi fitur data secara otomatis, memungkinkan sistem untuk mempelajari data secara berlapis, dari fitur yang paling mendasar hingga representasi yang lebih kompleks (LeCun dkk., 2015). Salah satu jenis model *deep learning* yang diterapkan dalam pemrosesan gambar dan video pada ElCue adalah *Convolutional Neural Network* (CNN).

2.3 Convolutional Neural Network (CNN)

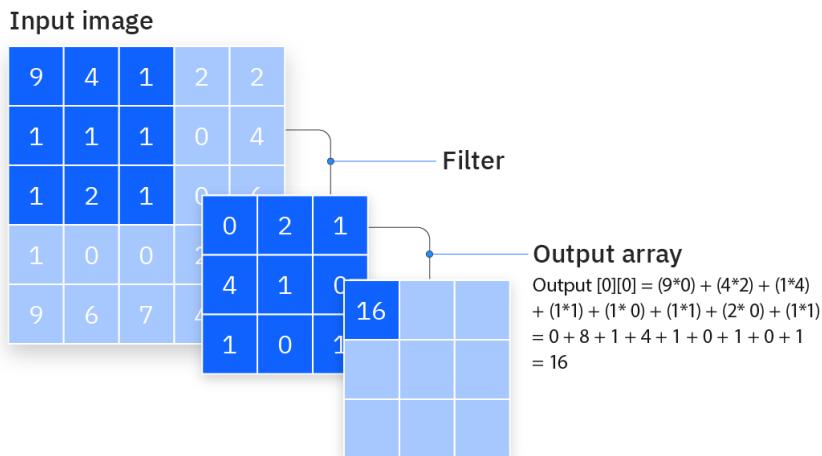
Convolutional Neural Network (CNN) adalah jenis jaringan saraf yang dirancang khusus untuk menangani data dengan struktur grid atau kisi, seperti halnya gambar dua dimensi. CNN merupakan arsitektur yang efektif dalam mengenali pola lokal pada data spasial, dengan mengandalkan operasi konvolusi untuk mengekstrak fitur-fitur penting secara bertahap (LeCun dkk., 1998). CNN menggunakan *convolutional layer*, *pooling layer*, dan *fully connected layer* untuk menganalisis dan mempelajari representasi data, yang memungkinkan sistem ini untuk mengenali objek dalam gambar dengan tingkat akurasi yang tinggi.



Gambar 2.2 Arsitektur CNN secara Sederhana (Sumber: Phung & Rhee, 2019)

2.3.1 Lapisan Konvolusi (Convolutional Layer)

Lapisan konvolusi (*Convolutional Layer*) merupakan komponen utama dalam CNN yang bertugas untuk mengekstraksi fitur dari input, seperti gambar. Pada lapisan ini, operasi konvolusi dilakukan dengan cara menerapkan filter atau kernel pada input untuk menghasilkan fitur yang merepresentasikan informasi lokal (misalnya tepi, tekstur, atau pola) pada gambar. Operasi ini memungkinkan jaringan untuk mengenali pola-pola yang relevan di berbagai posisi dalam gambar. Lapisan konvolusi biasanya terdiri dari beberapa filter yang berbeda, yang memungkinkan jaringan untuk mengenali berbagai jenis fitur. Penggunaan filter dalam konvolusi memungkinkan jaringan untuk secara otomatis mempelajari representasi fitur yang lebih kompleks seiring dengan bertambahnya kedalaman jaringan (LeCun dkk., 1998).



Gambar 2.3 Proses Operasi pada *Convolutional Layer* (Sumber: IBM, 2023)

Konvolusi dilakukan dengan menerapkan sebuah filter (atau kernel) pada input image, yang merupakan kumpulan nilai piksel. Filter ini memiliki ukuran lebih kecil dibandingkan dengan citra asli dan bergerak secara bertahap melintasi gambar dengan cara *sliding window*. Pada setiap langkah, nilai dalam filter dikalikan dengan nilai piksel yang sesuai dalam area gambar yang sedang diproses. Hasil perkalian ini kemudian dijumlahkan untuk menghasilkan satu nilai dalam output array. Misalnya, pada gambar, perhitungan untuk posisi pertama dalam output array $[0][0]$ dilakukan dengan:

$$(9 \times 0) + (4 \times 2) + (1 \times 4) + (1 \times 1) + (1 \times 0) + (1 \times 1) + (2 \times 0) + (0 \times 1) + (1 \times 1) = 16$$

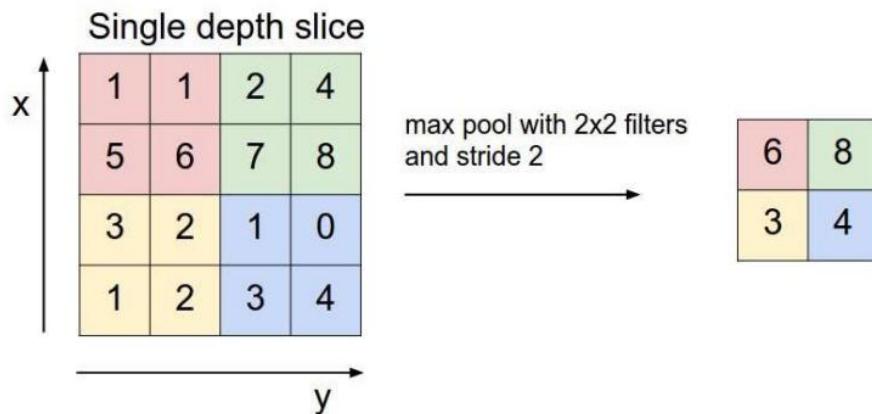
Proses ini berulang untuk seluruh posisi dalam gambar, menghasilkan representasi baru yang menyoroti fitur tertentu.

2.3.2 Pooling Layer

Setelah lapisan konvolusi, lapisan *pooling* diterapkan untuk mengurangi dimensi *output* dan meminimalkan informasi yang tidak penting. Proses *pooling* biasanya dilakukan dengan menggunakan operasi *Max Pooling* atau *Average Pooling*. Pada *Max Pooling*, sebuah kernel berukuran tertentu (misalnya 2×2) akan bergerak melintasi matriks input. Pada setiap langkah, nilai terbesar dari area yang dicakup oleh kernel akan dipilih dan dimasukkan ke dalam matriks keluaran. Sebaliknya, dalam *Average Pooling*, kernel dengan ukuran yang sama akan mengambil nilai rata-rata

dari semua elemen dalam area yang dicakup dan memasukkannya ke dalam matriks keluaran.

Proses *pooling* diterapkan pada setiap saluran dalam tensor input, yang menghasilkan tensor output dengan ukuran yang lebih kecil. Namun, perlu diingat bahwa lapisan *pooling* hanya mengurangi dimensi spasial (panjang dan lebar), sementara jumlah *channels* (kedalaman) tidak mengalami perubahan (Peryanto, dkk., 2020). Hal ini tidak hanya mengurangi jumlah parameter yang harus dipelajari, tetapi juga membantu meningkatkan ketahanan jaringan terhadap pergeseran dan perubahan kecil pada input. Proses ini juga berfungsi untuk mempercepat komputasi dan mengurangi risiko *overfitting*.



Gambar 2.4 Operasi Max Pooling (Sumber: Peryanto, dkk., 2020)

2.3.3 Fungsi Aktivasi (Activation Function)

Fungsi aktivasi digunakan untuk menambahkan elemen non-linear dalam jaringan, memungkinkan CNN untuk memodelkan hubungan yang lebih kompleks. Salah satu fungsi aktivasi yang paling sering digunakan dalam CNN adalah ReLU (*Rectified Linear Unit*), yang memiliki keunggulan dalam mengatasi masalah *vanishing gradient*, sehingga pelatihan jaringan menjadi lebih efisien. ReLU berfungsi dengan mengubah nilai negatif menjadi nol, sementara nilai positif tetap dipertahankan, sehingga hanya informasi yang relevan yang diteruskan ke lapisan selanjutnya.

$$f(x) = \max(0, x) \dots \dots \dots (1)$$

Artinya, untuk setiap nilai input x :

- Jika $x > 0$, maka output tetap x .

- Jika $x \leq 0$, maka output menjadi 0.

Selain itu, salah satu fungsi aktivasi yang umum digunakan adalah Softmax yang digunakan untuk *multiclass classification*. Softmax adalah fungsi aktivasi yang mengambil vektor z berisi nilai real dan mengonversinya menjadi vektor probabilitas yang jumlahnya selalu 1. Secara matematis, fungsi Softmax didefinisikan sebagai:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \dots \quad (2)$$

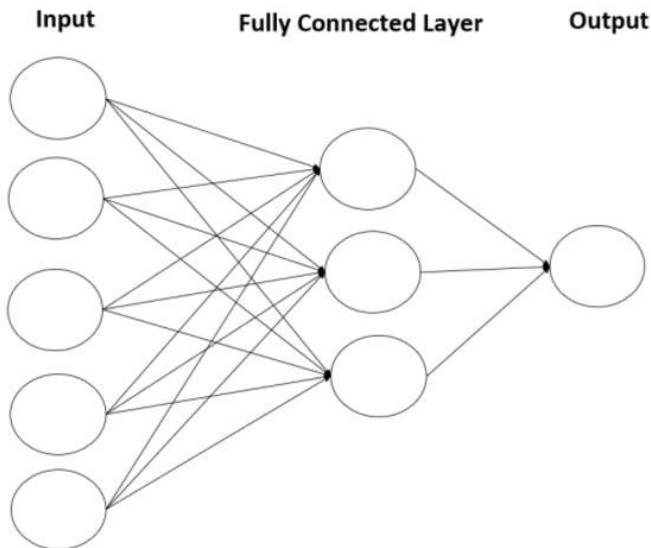
di mana:

- z_i adalah nilai input ke-i dari sekumpulan n nilai.
- e^{z_i} adalah eksponensial dari nilai input ke-i.
- $\sum_{j=1}^n e^{z_j}$ adalah jumlah dari semua eksponensial elemen dalam vektor z , yang memastikan bahwa hasil keluaran selalu membentuk distribusi probabilitas ($\sum \sigma(z) = 1$).

2.3.4 Fully Connected Layer

Setelah melewati beberapa lapisan konvolusi dan *pooling*, lapisan *fully connected* digunakan untuk menghubungkan seluruh neuron di lapisan sebelumnya. Pada lapisan ini, setiap neuron terhubung dengan neuron-neuron di lapisan berikutnya, memungkinkan jaringan untuk mengintegrasikan semua fitur yang telah dipelajari dan menghasilkan keputusan klasifikasi akhir. Lapisan *fully connected* untuk mentransformasikan ruang fitur agar masalah yang sedang diproses menjadi lebih mudah dipecahkan.

Dalam proses transformasi ini, jumlah dimensi dapat bertambah, berkurang, atau tetap sama. Pada setiap kasus, dimensi baru yang terbentuk merupakan kombinasi linear dari dimensi yang ada di lapisan sebelumnya. Kemudian, fungsi aktivasi digunakan untuk memberikan sifat non-linear pada dimensi baru tersebut. Pada ilustrasi yang ditunjukkan dalam Gambar 2.5, lapisan *fully connected* mengubah ruang fitur dari dimensi lima menjadi dimensi tiga (Kalayci & Asan, 2022).



Gambar 2.5 Fully Connected Layer (Sumber: Kalayci & Asan, 2022)

2.4 Pre-trained MobileNet

MobileNet merupakan arsitektur *deep learning* yang dikembangkan khusus untuk *vision application* pada perangkat *mobile* dan *embedded*, di mana sumber daya komputasi terbatas. Inovasi utama MobileNet adalah penggunaan *depthwise separable convolutions* untuk menggantikan operasi konvolusi standar. Dalam konvolusi standar, *feature map* input dan output keduanya dikonvolusi dengan seperangkat filter, yang mengakibatkan biaya komputasi yang tinggi, terutama ketika *feature map* input dan output memiliki jumlah saluran yang besar. Sebaliknya, *depthwise separable convolutions* memecah operasi ini menjadi dua langkah: *depthwise convolutions*, di mana setiap saluran input difilter secara independen, dan *pointwise convolutions*, di mana hasil konvolusi kedalaman digabungkan menggunakan konvolusi menggunakan 1×1 filter.

Proses faktorisasi ini secara signifikan mengurangi jumlah operasi dan parameter, sehingga model menjadi lebih efisien dalam hal komputasi sambil tetap mempertahankan tingkat akurasi yang tinggi. MobileNet juga memperkenalkan dua *hyperparameter*—*width multiplier* dan *resolution multiplier*—yang memungkinkan kontrol lebih lanjut terhadap *trade-off* antara akurasi, latensi, dan ukuran model, memungkinkan MobileNet untuk disesuaikan berdasarkan sumber daya perangkat yang tersedia (Howard dkk., 2017).

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Gambar 2.6 Arsitektur MobileNet (Sumber: Howard dkk., 2017)

MobileNetV2 membangun fondasi yang telah diletakkan oleh MobileNet, dengan memperkenalkan perbaikan signifikan untuk meningkatkan kinerja, terutama dalam hal efisiensi. Inovasi dari MobileNetV2 adalah *inverted residual blocks* dengan *bottleneck linier*. Dalam jaringan residu tradisional, input diteruskan melalui rangkaian lapisan, dan output digabungkan dengan input untuk membentuk output akhir. Namun, MobileNetV2 membalikkan struktur ini dengan memulai dari input berdimensi rendah, memperluasnya ke dimensi yang lebih tinggi, menerapkan konvolusi terpisah secara kedalaman untuk penyaringan fitur, dan kemudian mengompresi *feature map* kembali ke representasi berdimensi lebih rendah melalui *bottleneck linier*. Pendekatan ini memastikan bahwa fitur-fitur penting tetap terjaga sambil menjaga biaya komputasi tetap rendah. Selain itu, MobileNetV2 menghilangkan non-linearitas di lapisan *bottleneck*, yang membantu mencegah hilangnya fitur penting (Sandler dkk., 2019).

Input	Operator	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Gambar 2.7 Arsitektur MobileNetV2 (Sumber: Sandler dkk., 2019)

2.5 Fine-tuning untuk Optimasi Model CNN

Fine-tuning merupakan teknik dalam *transfer learning* yang bertujuan untuk menyesuaikan model yang telah dilatih sebelumnya agar dapat bekerja lebih optimal pada tugas atau domain baru. Metode ini memanfaatkan bobot model yang telah dipelajari dari dataset sumber yang besar, kemudian menyesuaikannya dengan dataset target yang umumnya lebih kecil. Li dkk. (2019) mendefinisikan *fine-tuning* sebagai cara efektif untuk mentransfer pengetahuan dari dataset sumber ke tugas target dengan memodifikasi parameter model sesuai dengan karakteristik data baru

Fine-tuning sering digunakan dalam berbagai aplikasi kecerdasan buatan, terutama ketika jumlah data pelatihan yang tersedia terbatas. Teknik ini memungkinkan model untuk mempertahankan informasi yang telah dipelajari sebelumnya sambil menyesuaikan fitur agar lebih relevan dengan tugas baru. Faktor seperti jumlah kategori pada dataset sumber dan kemiripan distribusi data sangat memengaruhi keberhasilan *fine-tuning* (Li dkk., 2019). Dalam skenario *out-of-distribution generalization*, *fine-tuning* dengan tingkat dropout tinggi dapat menghasilkan performa lebih baik dibandingkan metode *ensemble learning* dan *weight averaging* (Zhang & Bottou, 2024). Dengan demikian, *fine-tuning* menjadi strategi yang efisien dalam meningkatkan kinerja model tanpa perlu melatihnya dari awal.

2.5.1 Additional Layers

Menambahkan lapisan tambahan (*additional layers*) pada model deep learning bertujuan untuk memperkuat kemampuan model dalam mengenali pola atau fitur yang lebih kompleks. Lapisan tambahan ini sering kali berupa lapisan konvolusional (*convolutional layers*) yang dapat menangkap fitur yang lebih mendalam, atau lapisan *fully connected (dense layers)* yang digunakan untuk memetakan fitur yang telah diekstraksi ke dalam keputusan akhir. Penambahan lapisan ini memungkinkan model untuk meningkatkan kapasitas representasi dan menyesuaikan diri dengan tugas yang lebih rumit, seperti pengenalan objek yang memerlukan pemahaman tingkat lanjut atau variasi dalam data. Di samping itu, lapisan tambahan juga berfungsi untuk mengurangi risiko *underfitting* dengan memberi model kemampuan untuk mempelajari lebih banyak informasi yang relevan, serta membantu model dalam menangani data yang bervariasi, seperti kondisi pencahayaan atau sudut pandang yang berbeda (Fitroh & Uyun, 2023).

2.5.2 Regularization

Regularization merupakan teknik yang digunakan dalam *deep learning* dan *machine learning* untuk mencegah *overfitting*, yaitu kondisi di mana model terlalu fokus pada detail atau noise dalam data pelatihan, yang mengakibatkan penurunan kemampuan model untuk menggeneralisasi pada data yang belum pernah dilihat. *Overfitting* sering terjadi ketika model memiliki terlalu banyak parameter atau kapasitas untuk mempelajari hubungan yang sangat khusus dalam data pelatihan, sehingga model kesulitan dalam menghadapi data baru.

Tujuan dari *regularization* adalah untuk menambah penalti pada model agar lebih sederhana, menghindari kompleksitas berlebih, dan memaksa model untuk menemukan pola yang lebih umum dalam data. *Regularization* membantu menciptakan model yang lebih *robust*, yaitu model yang dapat bekerja dengan baik pada data baru meskipun data pelatihan mungkin tidak sempurna (IBM, 2023).

Ada beberapa metode *regularisasi* yang digunakan dalam *deep learning*, yang paling umum adalah L2 *Regularization (Weight Decay)*, L1

Regularization, Dropout, dan Early Stopping. Masing-masing metode ini memiliki pendekatan yang berbeda dalam membatasi kompleksitas model dan mengurangi risiko *overfitting*.

Dropout adalah teknik regularisasi yang lebih langsung, di mana selama pelatihan, neuron-neuron dalam jaringan secara acak dihilangkan (dimatikan) pada setiap iterasi. Ini mencegah model terlalu bergantung pada beberapa neuron tertentu dan memaksa jaringan untuk mempelajari representasi yang lebih *robust*. Dengan cara ini, dropout memperkenalkan bentuk *ensemble learning*, di mana berbagai subset dari model (dengan neuron yang berbeda) berfungsi secara independen, yang meningkatkan kemampuan model untuk menggeneralisasi.

2.5.3 Loss Function

Loss function sangat penting karena berfungsi sebagai pengukur seberapa besar perbedaan antara prediksi model dan nilai sebenarnya. Salah satu tipe fungsi loss yang umum diterapkan adalah *crossentropy loss*, yang juga dikenal sebagai *log loss*. Fungsi ini sangat berguna untuk mengevaluasi kinerja model klasifikasi yang menghasilkan output dalam bentuk probabilitas dengan nilai antara 0 dan 1. Nilai dari *crossentropy loss* akan semakin besar jika probabilitas yang diprediksi model semakin menyimpang dari *actual label* (Liew dkk., 2021).

Crossentropy loss berasal dari konsep teori informasi, yang menghitung selisih antara dua distribusi probabilitas untuk variabel acak dalam peristiwa tertentu. Dalam konteks machine learning, fungsi ini dipakai untuk mengukur selisih antara distribusi probabilitas yang sebenarnya dan yang diprediksi oleh model. Untuk klasifikasi biner, rumus dari binary cross-entropy loss dapat dituliskan sebagai berikut:

$$L_{BCE}(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \dots \dots \dots \quad (3)$$

di mana y adalah label kelas aktual yang bernilai 0 atau 1, dan \hat{y} adalah probabilitas prediksi oleh model bahwa sampel termasuk kelas 1.

Dalam kasus *multiclass classification*, *loss function* yang digunakan adalah *categorical cross-entropy*. Fungsi ini diperluas untuk menangani

lebih dari dua kelas. *Categorical cross-entropy* menghitung loss dengan merata-ratakan loss yang dihasilkan dari setiap kelas, menggunakan rumus:

$$L_{CCE} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{ic} \log(p_{ic}) \dots \quad (4)$$

di mana N adalah jumlah sampel, C adalah jumlah kelas, y_{ic} adalah indikator (0 atau 1) jika kelas C adalah kelas yang benar untuk sampel i , dan p_{ic} adalah probabilitas prediksi model bahwa sampel i termasuk dalam kelas C . Fungsi ini efektif untuk menangani situasi di mana setiap sampel bisa dikategorikan ke dalam satu dari beberapa kelas, dan mencerminkan seberapa baik model melakukan klasifikasi sampel tersebut ke dalam kelas yang benar.

Categorical cross-entropy sangat relevan dalam aplikasi nyata di mana ketepatan klasifikasi setiap kelas penting, seperti dalam segmentasi gambar dan pengenalan objek, karena memungkinkan evaluasi yang lebih spesifik terhadap efisiensi model dalam mengklasifikasikan setiap kelas dengan benar (Yeung dkk., 2022).

2.5.3 Learning Rate

Learning rate adalah *hyperparameter* dalam *machine learning* yang mengatur besarnya perubahan bobot model selama proses pembaruan parameter menggunakan algoritma optimasi. *Learning rate* memiliki peran yang krusial dalam mengatur kecepatan model dalam mempelajari data yang diberikan. Apabila nilai *learning rate* terlalu rendah, proses konvergensi akan terjadi sangat lambat, yang menyebabkan model membutuhkan lebih banyak iterasi untuk menemukan solusi yang optimal. Sebaliknya, jika *learning rate* terlalu besar, model dapat mengalami ketidakstabilan dan gagal mencapai konvergensi karena pembaruan parameter yang terlalu drastis.

Dalam praktiknya, terdapat berbagai pendekatan dalam menentukan *learning rate* yang optimal, seperti penggunaan *fixed learning rate*, *adaptive learning rate*, dan *cyclical learning rate*. Pendekatan *fixed learning rate* menggunakan nilai tetap sepanjang pelatihan, sedangkan *adaptive learning rate* menyesuaikan nilai *learning rate* berdasarkan perubahan gradien atau magnitudo parameter model. Sementara itu, *cyclical learning rate*

memungkinkan nilai *learning rate* berfluktuasi dalam rentang tertentu selama pelatihan, yang dapat membantu model untuk mencapai konvergensi lebih cepat dan menghindari perangkap *local minima*. Pemilihan strategi *learning rate* yang tepat sangat penting untuk memastikan model mencapai performa terbaik dengan efisiensi yang optimal (Smith, 2017).

2.5.4 Optimizer

Optimizer merupakan algoritma yang digunakan untuk memperbarui parameter model dengan mengacu pada gradien yang dihitung dari *loss function*. Tugas utama *optimizer* adalah menentukan metode pembaruan bobot pada setiap iterasi selama proses pelatihan. Beberapa *optimizer*, seperti Stochastic Gradient Descent (SGD), Adam, atau RMSProp, memiliki cara yang berbeda dalam menghitung pembaruan bobot dan mengatur langkah-langkah yang diambil dalam proses optimasi.

Optimizers mengatur bagaimana gradien digunakan untuk memperbarui bobot, termasuk apakah gradien dihitung hanya berdasarkan sampel acak (*stochastic*) atau menggunakan seluruh dataset (*batch*). Beberapa *optimizer* juga menggunakan strategi tambahan, seperti momentum untuk mempercepat konvergensi atau adaptasi *learning rate* untuk menyesuaikan ukuran langkah berdasarkan pola gradien. Dalam beberapa *optimizer* modern seperti Adam (*Adaptive Moment Estimation*), *learning rate* berfungsi sebagai parameter utama yang mengontrol laju pembaruan, tetapi optimizer tersebut juga memanfaatkan informasi tambahan seperti rata-rata momentum (untuk mempercepat proses pelatihan) dan skala variansi gradien (untuk menyesuaikan pembaruan berdasarkan distribusi gradien).

Adam merupakan salah satu *optimizer* yang paling sering digunakan dalam pelatihan model *deep learning*. Adam merupakan kombinasi dari dua pendekatan optimasi yang sudah ada sebelumnya, Momentum dan RMSProp. Adam dirancang untuk menggabungkan keuntungan kedua metode ini, yaitu kecepatan konvergensi dari momentum dan kemampuan untuk menyesuaikan *learning rate* dari RMSProp, sehingga membuatnya lebih efisien dan lebih stabil dalam berbagai skenario pelatihan. Adam

mengadaptasi *learning rate* untuk setiap parameter dengan mempertimbangkan pertama-tama rata-rata momentum (*mean*) dari gradien dan kedua variansi (*uncentered variance*) dari gradien tersebut. Dengan cara ini, Adam dapat menyesuaikan langkah pembaruan untuk setiap parameter secara dinamis, yang sangat berguna terutama pada model dengan banyak parameter dan data yang sangat besar (Kingma & Ba, 2014).

2.6 Evaluasi Model

Proses evaluasi model adalah langkah penting dalam pengembangan sistem *machine learning*, khususnya untuk model yang digunakan dalam tugas klasifikasi citra atau video. Tujuan dari evaluasi adalah untuk menilai seberapa efektif model yang telah dioptimalkan dalam mendekripsi. Berikut adalah beberapa metrik evaluasi yang digunakan untuk mengevaluasi kinerja model klasifikasi.

2.6.1 Metrik Evaluasi Model

Metrik evaluasi berfungsi untuk mengukur kualitas dan akurasi model klasifikasi yang telah dilatih. Metrik-metrik ini berperan dalam memahami sejauh mana model mampu memprediksi data yang diberikan. Berikut adalah beberapa metrik evaluasi yang diterapkan dalam penelitian ini:

A. Akurasi (*Accuracy*)

Akurasi merupakan metrik yang paling sering digunakan untuk menilai kinerja model dalam tugas klasifikasi. Metrik ini menghitung rasio antara jumlah prediksi yang benar dengan total data yang ada. Secara matematis, akurasi dihitung dengan rumus:

$$\text{Accuracy} = \frac{TP/TN}{P+N} \dots\dots\dots (5)$$

Dimana:

- TP: True Positive (jumlah prediksi positif yang benar)
- TN: True Negative (jumlah prediksi negatif yang benar)
- P: Total positif (TP + FN)
- N: Total negatif (TN + FP)

B. Presisi (*Precision*)

Precision mengukur seberapa akurat model dalam memprediksi kelas positif, yaitu apakah gestur bahasa isyarat yang diprediksi benar-benar

sesuai dengan kelas yang diharapkan. *Precision* sangat penting ketika biaya kesalahan dalam memprediksi kelas positif adalah tinggi, seperti ketika model salah mengidentifikasi gestur yang tidak relevan sebagai gestur yang benar. Rumus untuk menghitung presisi adalah:

$$\text{Precision} = \frac{TP}{TP+FP} \dots\dots\dots (6)$$

Dimana:

- TP: True Positive
- FP: False Positive

C. *Recall*

Recall menentukan kemampuan model untuk menemukan kelas positif yang akurat. Dalam konteks deteksi objek, recall menggambarkan proporsi objek yang berhasil dikenali oleh model. Metrik ini krusial untuk memastikan bahwa model tidak mengabaikan kelas positif, meskipun beberapa prediksi dapat salah. Rumus untuk recall adalah:

$$\text{Recall} = \frac{TP}{TP+FN} \dots\dots\dots (7)$$

Dimana:

- TP: True Positive
- FN: False Negative

D. F1-Score

F1-Score memberikan representasi yang lebih seimbang antara *precision* dan *recall* karena merupakan rata-rata harmonik antara keduanya. F1-Score sangat bermanfaat ketika terdapat ketidakseimbangan antara *precision* dan *recall*, serta saat model perlu mencapai keseimbangan antara keduanya. F1-Score dihitung dengan rumus:

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \dots\dots\dots (8)$$

2.6.2 Confusion Matrix

Tabel *Confusion Matrix* menunjukkan bagaimana model klasifikasi bekerja dengan memberikan detail tentang jumlah prediksi yang tepat dan keliru untuk setiap kelas. Matriks ini memberikan wawasan mengenai cara model

mengklasifikasikan data ke dalam masing-masing kategori. Secara umum, *Confusion matrix* pada umumnya memiliki format berikut (Arjaria dkk., 2021):

	Prediksi Gestur Benar	Prediksi Gestur Salah
Gestur Benar	True Positives (TP)	False Negatives (FN)
Gestur Salah	False Positives (FP)	True Negatives (TN)

Confusion matrix sangat bermanfaat untuk menemukan masalah potensial, seperti ketidakseimbangan data atau bias model terhadap kelas tertentu. Ini memungkinkan penghitungan berbagai metrik, termasuk *precision*, *recall*, dan *F1-Score*, untuk menilai kinerja model secara menyeluruh (Raschka, 2018).

2.7 Penelitian Terkait

Terdapat beberapa referensi penelitian yang relevan dengan penggunaan *deep learning* dan MobileNetV2 dalam konteks pengenalan bahasa isyarat. Salah satunya adalah penelitian yang dilakukan oleh Shania dkk (2022) yaitu mengembangkan sistem penerjemah bahasa isyarat Indonesia (BISINDO) berbasis video menggunakan Convolutional Neural Network (CNN) dengan *transfer learning*. Penelitian ini menggunakan model *pre-trained* seperti MobileNetV2, ResNet50V2, dan Xception untuk mengklasifikasikan 11 kosakata BISINDO yang sering digunakan. Hasil eksperimen menunjukkan bahwa model Xception memberikan F1-Score terbaik sebesar 98,5% dalam klasifikasi gerakan tangan dalam BISINDO. Penelitian ini menyoroti efisiensi penggunaan *transfer learning* dalam penerjemah bahasa isyarat berbasis video dengan menggunakan *pre-trained* model CNN, serta menunjukkan potensi MobileNetV2 dalam aplikasi *real-time*.

Penelitian lain yang sejalan dengan tema ini dilakukan oleh Wijaya dkk (2024) yaitu mengembangkan sistem *sign language translator* menggunakan model *Artificial Neural Network* (ANN) untuk menerjemahkan bahasa isyarat SIBI ke dalam bahasa Indonesia. Penelitian ini melakukan analisis kuantitatif untuk mengevaluasi akurasi model dalam memprediksi alfabet dan kata-kata. Hasil eksperimen menunjukkan bahwa model ANN berhasil mencapai akurasi 96,15% untuk prediksi alfabet dan 99,45% untuk prediksi kata dalam waktu prediksi yang

tidak melebihi 0,15 detik. Hasil tersebut mengindikasikan bahwa model ANN cocok untuk diterapkan sebagai sistem penerjemah bahasa isyarat Indonesia yang efisien dalam hal waktu dan akurasi.

Penelitian lain yang dilakukan oleh Aldhahri dkk (2022) yaitu mengembangkan sistem pengenalan Bahasa Isyarat Arab (ArSL) dengan menggunakan Convolutional Neural Network (CNN) yang dioptimalkan dengan model MobileNet. Dataset yang digunakan adalah Arabic Alphabets Sign Language Dataset (ArASL2018), yang berisi gambar tanda tangan alfabet bahasa Arab. Hasil penelitian menunjukkan akurasi pengenalan sebesar 94,46%, yang mengungguli model-model sebelumnya. Penelitian ini menekankan pada penggunaan MobileNet dalam pengenalan bahasa isyarat, dengan hasil yang baik meskipun menggunakan dataset terbatas. MobileNet menunjukkan kemampuannya dalam pengenalan tanda-tangan bahasa isyarat Arab dengan kecepatan dan akurasi yang efisien.

Selain itu, Pu dan Yi (2024) melakukan studi perbandingan kinerja antara MobileNet dan EfficientNet pada deteksi alfabet bahasa isyarat Amerika (ASL) menggunakan dataset kecil. Penelitian ini menguji kinerja kedua arsitektur pada 5 epoch, 10 epoch, dan 20 epoch, dengan hasil yang menunjukkan bahwa MobileNetV2 unggul, terutama pada 5 epoch, dalam hal akurasi dan penggunaan sumber daya yang lebih rendah. Penelitian ini juga menunjukkan bahwa MobileNetV2 lebih efisien dalam menangani dataset kecil dengan meminimalkan *overfitting*, menjadikannya pilihan yang sangat baik untuk *real-time application* pada perangkat dengan sumber daya terbatas.

Terakhir, Tamkar dkk (2024) mengembangkan sistem pengenalan Bahasa Isyarat India (ISL) secara *real-time* menggunakan MobileNetV2 dan transfer learning. Sistem ini memanfaatkan keunggulan MobileNetV2 yang efisien untuk pengenalan gerakan tangan ISL dengan latensi rendah, cocok untuk aplikasi pada perangkat *mobile* atau *embedded* dengan sumber daya terbatas. Dalam penelitian ini, MobileNetV2 dilatih menggunakan dataset ISL yang dikustomisasi dengan menambahkan lapisan khusus pada model yang sudah dilatih sebelumnya. Hasilnya menunjukkan akurasi yang tinggi dengan kemampuan *real-time* yang memadai untuk aplikasi sehari-hari. Penelitian ini menekankan pentingnya penggunaan

transfer learning untuk meningkatkan akurasi pengenalan bahasa isyarat pada perangkat terbatas.



BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem dilakukan untuk memahami berbagai aspek yang berkaitan dengan pengembangan model pendekripsi Bahasa Isyarat Indonesia (BISINDO). Tahap ini mencakup analisis masalah, analisis data, analisis kebutuhan dan diagram umum sistem.

3.1.1 Analisis Masalah

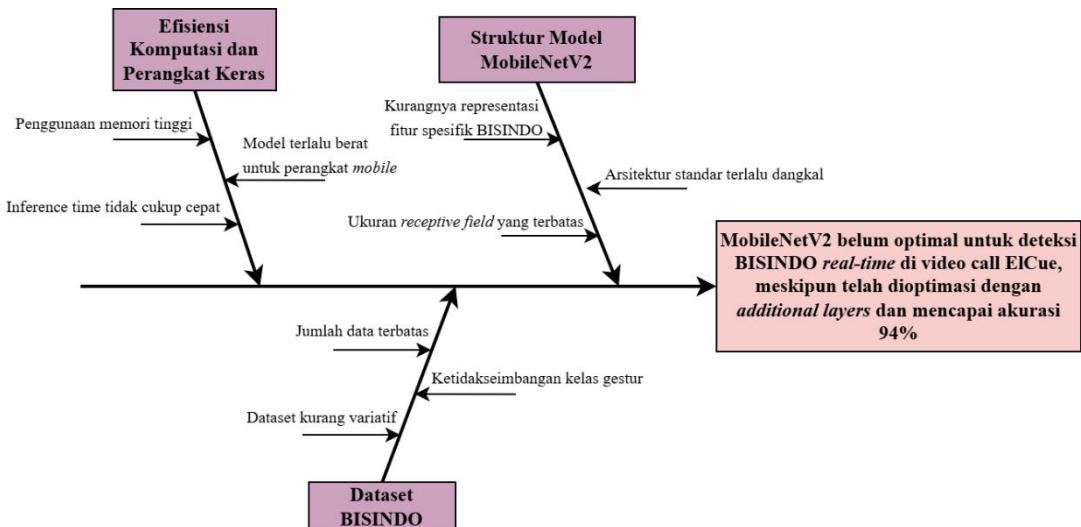
Pada penelitian sebelumnya yang telah dilakukan pada ElCue (2024), model MobileNetV2 dengan *additional layers* telah diterapkan untuk pendekripsi Bahasa Isyarat Indonesia (BISINDO) secara *real-time* dalam aplikasi *video call*. Model ini berhasil mencapai akurasi 94%, namun masih memiliki beberapa keterbatasan yang menghambat performa deteksi dalam lingkungannya.

Salah satu kendala utama adalah keterbatasan arsitektur MobileNetV2 yang masih kurang optimal dalam mengenali gestur BISINDO dari berbagai sudut pandang dan kondisi pencahayaan. *Receptive field* yang terbatas membuat model sulit menangkap fitur spesifik dari gerakan tangan yang kompleks, sehingga masih terjadi kesalahan klasifikasi, terutama pada gestur yang memiliki kemiripan bentuk. Selain itu, keterbatasan dataset yang tidak seimbang menyebabkan model kesulitan beradaptasi dengan variasi gestur dari pengguna yang berbeda.

Di sisi lain, dari segi efisiensi komputasi, meskipun MobileNetV2 dirancang sebagai model ringan, implementasinya dalam sistem *real-time* masih memiliki tantangan dalam hal konsumsi memori dan kecepatan inferensi, terutama ketika dijalankan pada perangkat mobile dengan spesifikasi terbatas. Hal ini menghambat pengalaman pengguna dalam aplikasi *video call* yang membutuhkan respons cepat dan akurasi tinggi.

Oleh karena itu, penelitian ini berfokus pada optimasi MobileNetV2 dengan teknik *fine-tuning* dan penambahan *additional layers* yang lebih efektif. Pendekatan ini bertujuan untuk meningkatkan representasi fitur

gesture BISINDO, meningkatkan akurasi deteksi melebihi hasil penelitian sebelumnya, serta memastikan performa model yang lebih efisien dan cepat dalam sistem *video call* ElCue secara *real-time*.



Gambar 3.1 Diagram Ishikawa

3.1.2 Analisis Data

Dalam penelitian ini, data yang digunakan berupa dataset Bahasa Isyarat Indonesia (BISINDO) yang dikumpulkan secara langsung bersama Gerakan Kesejahteraan untuk Tunarungu di Indonesia (GERKATIN) menggunakan program *data collection* berbasis *computer vision* yang dikembangkan secara mandiri. Dataset yang dikumpulkan terdiri dari 9 kelas yaitu “A”, “Asal”, “B”, “C”, “D”, “E”, “Halo”, “I”, “Terima Kasih” yang masing-masing merepresentasikan gestur dengan makna tertentu dalam BISINDO.

Struktur data yang digunakan meliputi format gambar JPEG dengan resolusi 128x128 pixel yang disesuaikan agar kompatibel dengan model MobileNetV2. Setiap data diklasifikasikan berdasarkan label gestur yang mewakili makna dalam BISINDO, dengan jumlah kelas isyarat yang ditentukan sesuai cakupan penelitian ini. Jumlah sampel untuk setiap kelas adalah lebih kurang 1000 gambar, sehingga total dataset yang dikumpulkan berjumlah 9488 gambar. Seluruh dataset disimpan di platform Kaggle agar dapat diakses dengan mudah.

Sebelum digunakan untuk pelatihan model, dataset akan melalui tahap *preprocessing* guna meningkatkan kualitas dan akurasi deteksi. Teknik yang digunakan meliputi augmentasi data, seperti *rotation*,

translation, zoom, dan flipping serta *edge detection*, untuk meningkatkan keberagaman sampel gestur BISINDO. Selain itu, dilakukan normalisasi nilai piksel agar distribusi data lebih seragam serta konversi gambar dari *grayscale* ke *RGB* agar kompatibel dengan arsitektur MobileNetV2.



Gambar 3.2 Dataset BISINDO

3.1.3 Analisis Kebutuhan

Dalam pengembangan sistem deteksi Bahasa Isyarat Indonesia (BISINDO) berbasis *deep learning*, analisis kebutuhan menjadi tahap penting untuk memastikan bahwa sistem yang dirancang dapat bekerja secara optimal dalam lingkungan *real-time*, seperti dalam aplikasi *video call*. Analisis ini mencakup identifikasi perangkat lunak dan perangkat keras yang

dibutuhkan untuk mendukung proses pelatihan serta penerapan model *deep learning* agar dapat beroperasi dengan efisien.

A. Analisis Kebutuhan Perangkat Lunak

Berikut ini adalah spesifikasi *cloud system* Google Colaboratory yang akan digunakan dalam penelitian ini:

Tabel 3.1 Spesifikasi Perangkat Lunak

Spesifikasi	Keterangan
Python Version	3.11.11
Tensorflow Version	2.18.0
RAM	12.67 GB
Disk	108 GB
TPU	-

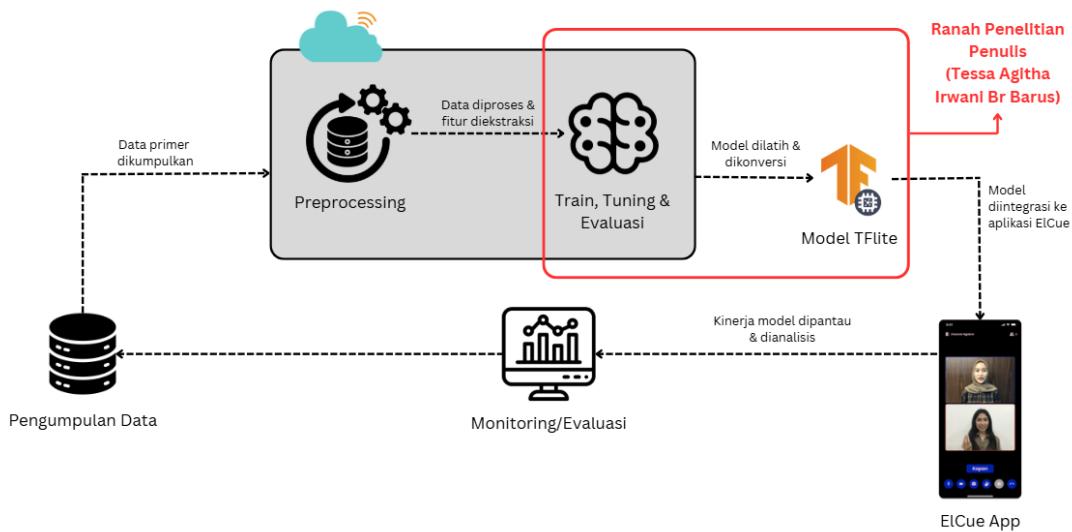
B. Analisis Kebutuhan Perangkat Keras

Spesifikasi *hardware* laptop pribadi yang digunakan dalam penelitian ini:

Tabel 3.2 Spesifikasi Perangkat Keras

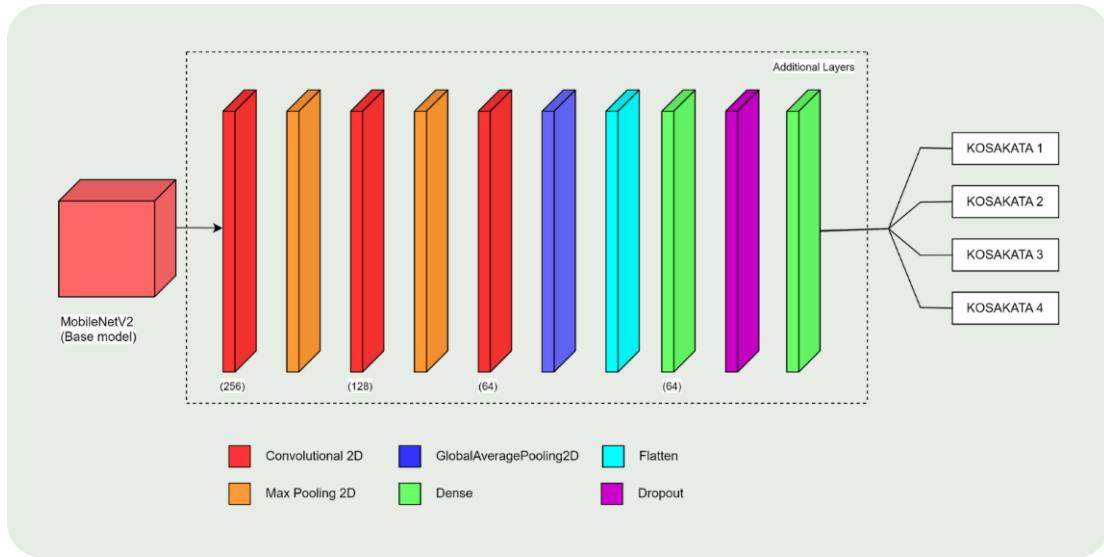
Spesifikasi	Keterangan
Processor	11th Gen Intel® Core™ i5-1135G7 @ 2.40GHz (8 CPUs), ~2.4GHz
RAM	8 GB
Storage	512 GB
Graphic Card	Intel® Iris® Ce Graphics
Operating System	Windows 11

3.1.4 Diagram Umum Sistem



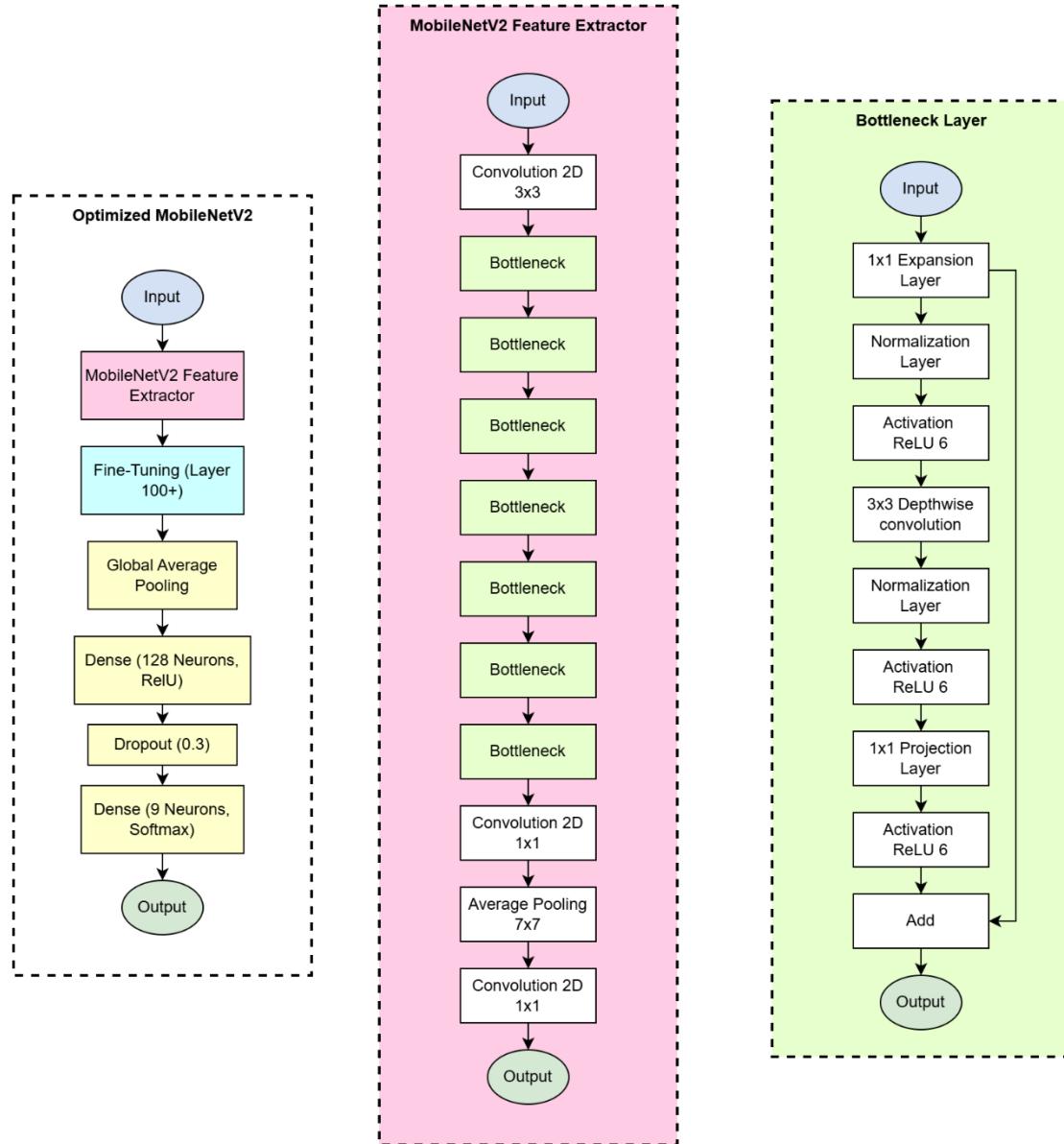
Gambar 3.3 Diagram Umum Sistem

Pada sistem yang dikembangkan sesuai dengan Gambar 3.3, proses pengolahan dan implementasi model deteksi Bahasa Isyarat Indonesia (BISINDO) berbasis *deep learning* yang terdiri dari beberapa tahapan utama. Tahapan ini mencakup pengumpulan dan pemrosesan data, pengembangan model, integrasi ke dalam sistem, serta pemantauan kinerja model. Fokus utama dalam penelitian ini adalah pada tahap pengembangan model (*Develop Model*), yang mencakup proses pelatihan, optimasi, dan evaluasi model *deep learning* agar dapat mengenali gestur BISINDO secara akurat dalam lingkungan *real-time*.



Gambar 3.4 Arsitektur Model Deteksi BISINDO ElCue dengan MobileNetV2
dan Additional Layers

Pada pengembangan model pendekripsi BISINDO aplikasi ElCue sebelumnya (2024) hanya menggunakan additional layers seperti Convolutional2D, MaxPooling2D, GlobalAveragePooling2D, Dense, Flatten, dan Dropout. Namun, pada tahap pengembangan model untuk penelitian ini, MobileNetV2 digunakan sebagai *feature extractor*, dengan memanfaatkan bobot *pre-trained* yang telah dilatih pada dataset ImageNet. Untuk meningkatkan kinerja model dalam mengenali gestur BISINDO, dilakukan *fine-tuning* dengan membuka kembali beberapa lapisan akhir MobileNetV2 setelah lapisan ke-100, sehingga model dapat menyesuaikan bobotnya terhadap karakteristik spesifik gestur yang terdapat dalam dataset BISINDO. Agar model dapat beradaptasi dengan baik terhadap data baru tanpa mengalami *overfitting*, *learning rate* diturunkan setelah *unfreezing* dilakukan, sehingga pembaruan bobot berlangsung secara bertahap dan stabil.



Gambar 3.5 Arsitektur Model Deteksi BISINDO ElCue dengan MobileNetV2 yang telah dioptimasi dengan *fine-tuning* dan *additional layers*

Selain *fine-tuning*, model dikembangkan dengan menambahkan lapisan tambahan (*additional layers*) untuk meningkatkan kapasitas representasi fitur dan klasifikasi. Lapisan ini terdiri dari *Global Average Pooling*, yang berfungsi untuk mereduksi dimensi fitur sebelum memasuki *fully connected layer*, diikuti dengan *Dense Layer* (128 neurons, ReLU) yang berfungsi sebagai lapisan klasifikasi utama. Untuk mengurangi risiko *overfitting*, diterapkan *Dropout* (0.3) sebelum memasuki *Output Layer*.

(Softmax, 9 kelas BISINDO), yang bertugas mengklasifikasikan gestur ke dalam salah satu dari sembilan kategori yang telah ditentukan.

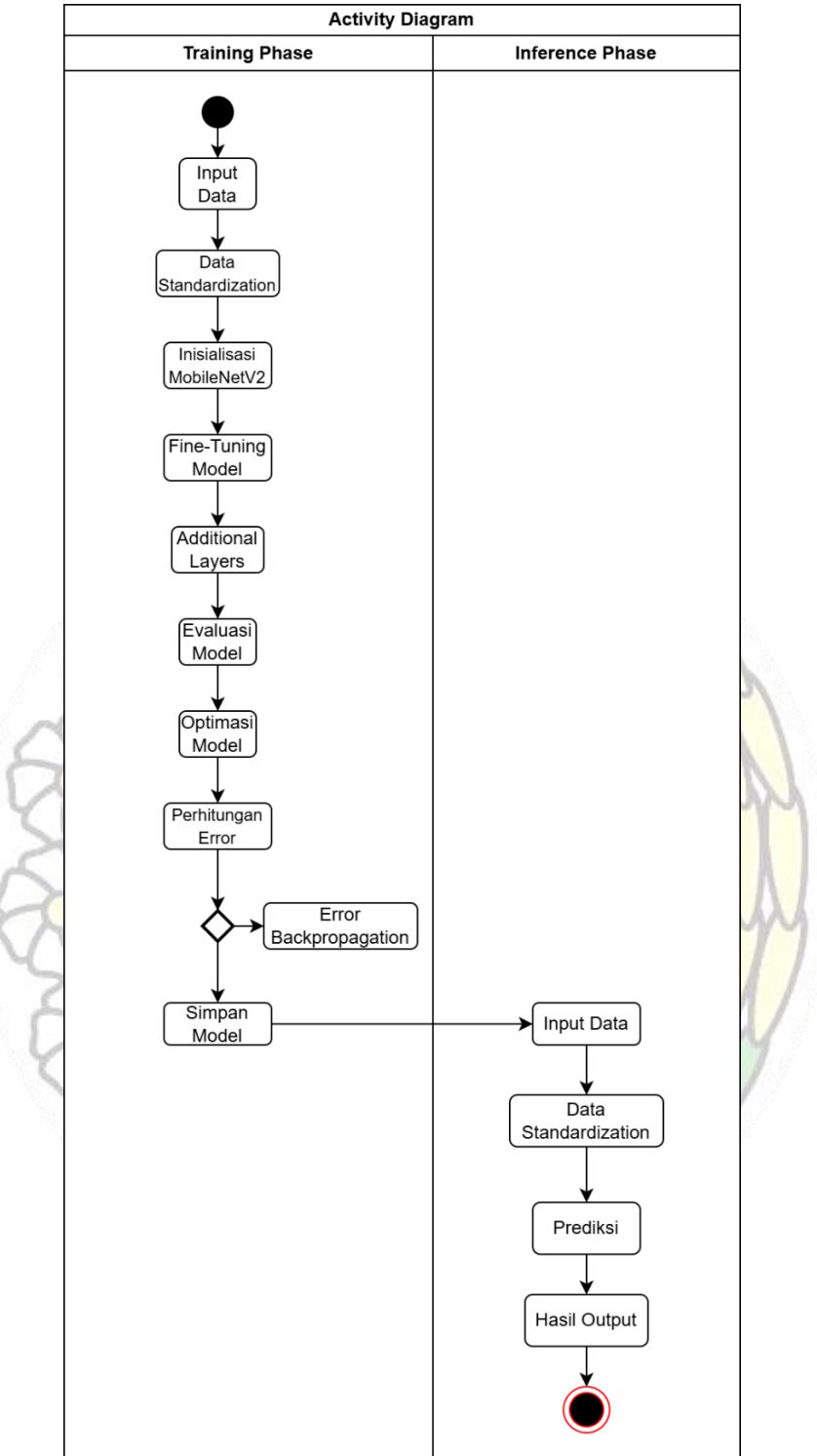
Model yang telah dibangun kemudian dievaluasi menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* untuk memastikan bahwa model dapat mengenali gestur BISINDO dengan akurasi yang optimal. Setelah model mencapai performa optimal, model tersebut akan diintegrasikan ke dalam sistem *video call* ElCue, sehingga dapat digunakan untuk mendeteksi dan menerjemahkan Bahasa Isyarat Indonesia secara *real-time*. Melalui pendekatan ini, sistem yang dirancang diharapkan dapat menawarkan solusi efisien, adaptif, dan tepat dalam mendeteksi gestur BISINDO di dunia nyata.

3.2 Pemodelan Sistem

3.2.1 Activity Diagram

Gambar 3.5 menggambarkan diagram *activity* dalam pengembangan dan penerapan model MobileNetV2 untuk pendeksi BISINDO, yang terdiri dari dua tahap utama, yaitu *Training Phase* dan *Inference Phase*. Pada tahap *Training Phase*, proses dimulai dengan input data yang kemudian distandarisasi (*Data Standardization*) sebelum digunakan untuk inisialisasi model MobileNetV2. Selanjutnya, model menjalani proses *fine-tuning* dengan penyesuaian bobot dan penambahan lapisan tambahan (*Additional Layers*) untuk meningkatkan akurasi klasifikasi.

Model kemudian dievaluasi berdasarkan metrik evaluasi, diikuti dengan optimasi model guna mengurangi kesalahan prediksi. Jika error masih tinggi, dilakukan perhitungan ulang error dan *backpropagation* untuk memperbarui bobot model, sebelum akhirnya model disimpan untuk tahap inferensi. Pada *Inference Phase*, data baru dimasukkan dan melalui proses standarisasi agar sesuai dengan format data latih, kemudian digunakan untuk prediksi (inferensi model). Hasil prediksi yang telah diproses dikembalikan dalam bentuk output yang merepresentasikan gestur BISINDO yang terdeteksi. Diagram ini menunjukkan bagaimana model dilatih, dioptimalkan, dan diterapkan untuk melakukan deteksi bahasa isyarat secara *real-time*.



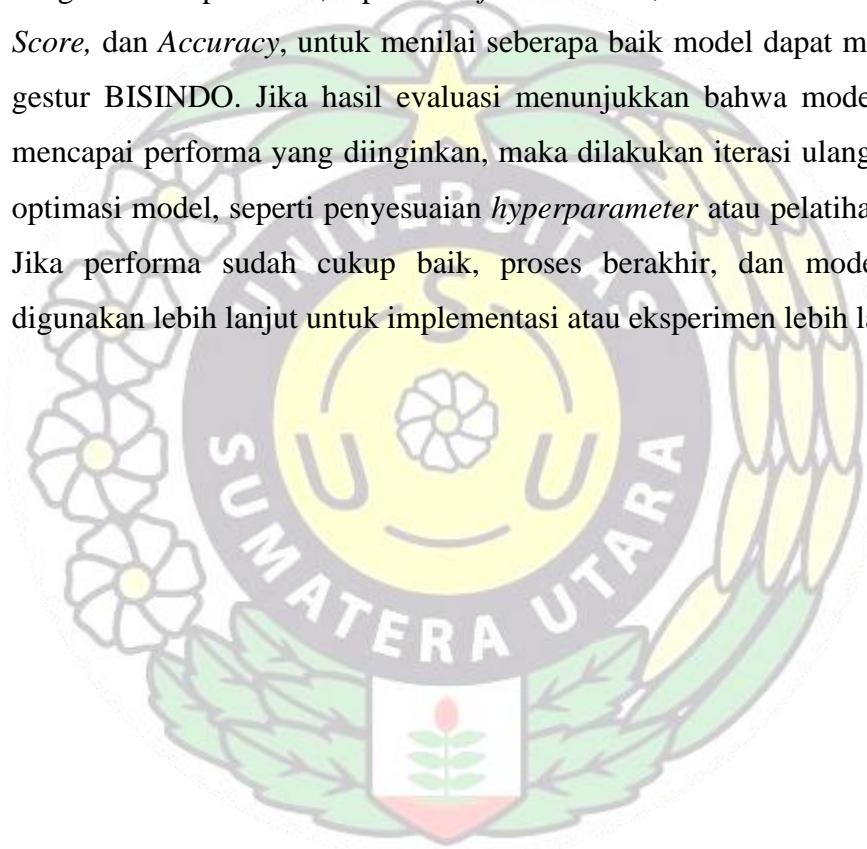
Gambar 3.6 Activity Diagram

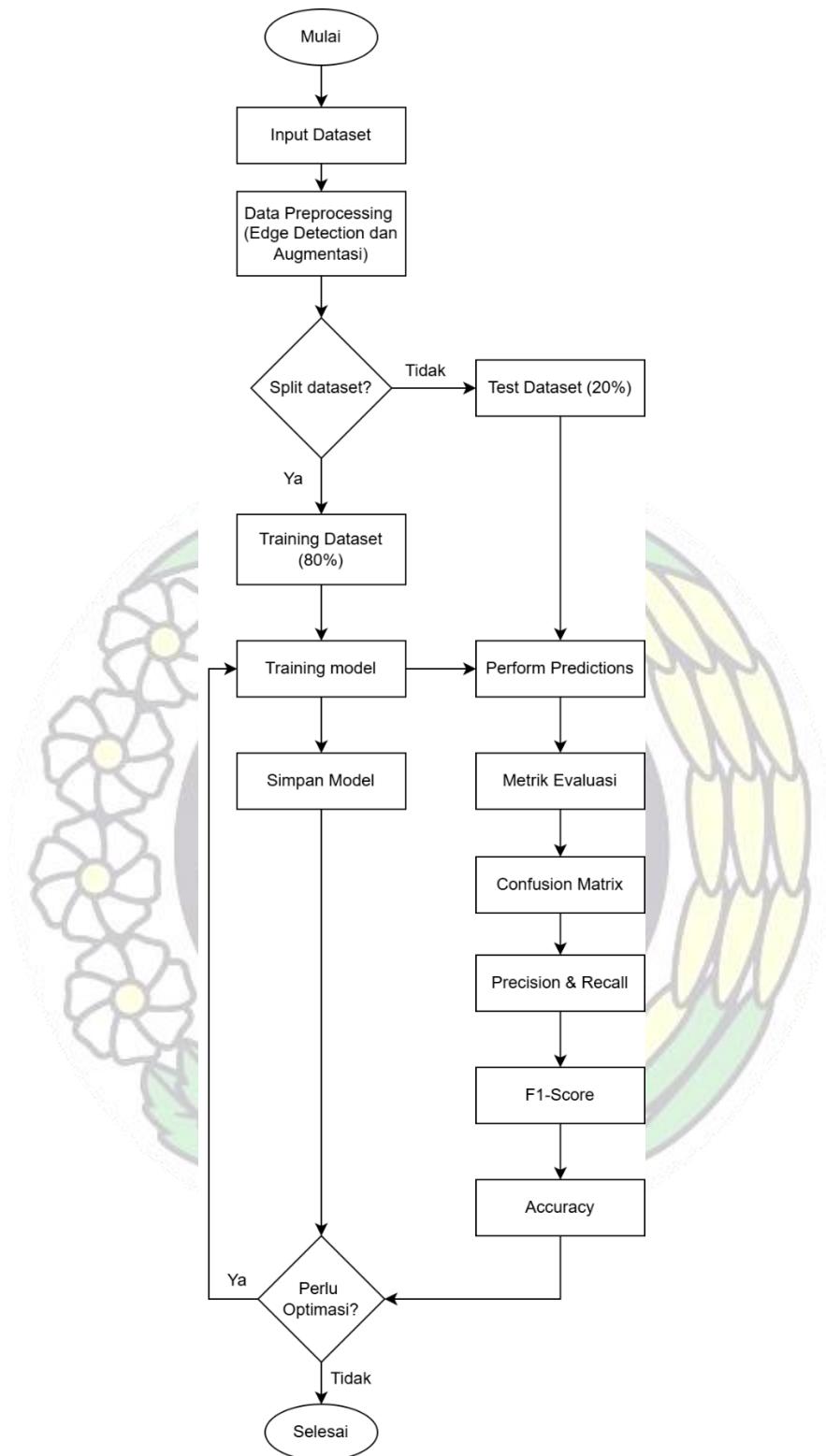
3.2.2 Flowchart

Gambar 3.6 menggambarkan *flowchart* pemrosesan data, pelatihan model, dan evaluasi performa dalam pengembangan model pendekripsi Bahasa Isyarat Indonesia (BISINDO) berbasis MobileNetV2. Proses dimulai

dengan input dataset, yang kemudian melalui tahap *preprocessing*, termasuk normalisasi, *edge detection*, dan augmentasi untuk memastikan data dalam kondisi optimal sebelum digunakan. Setelah dilakukan preprocessing, dataset dibagi menjadi 80% data digunakan untuk pelatihan dan 20% untuk pengujian, dimana data pelatihan digunakan untuk melatih model, dan data pengujian digunakan untuk mengevaluasi kinerja model setelah pelatihan.

Model yang telah dilatih akan disimpan dan dapat digunakan kembali untuk memprediksi data uji. Hasil prediksi kemudian dievaluasi dengan metrik performa, seperti *Confusion Matrix*, *Precision & Recall*, *F1-Score*, dan *Accuracy*, untuk menilai seberapa baik model dapat mengenali gestur BISINDO. Jika hasil evaluasi menunjukkan bahwa model belum mencapai performa yang diinginkan, maka dilakukan iterasi ulang dengan optimasi model, seperti penyesuaian *hyperparameter* atau pelatihan ulang. Jika performa sudah cukup baik, proses berakhir, dan model dapat digunakan lebih lanjut untuk implementasi atau eksperimen lebih lanjut.





Gambar 3.7 Flowchart

BAB 4

IMPLEMENTASI DAN PENGUJIAN

Bagian ini membahas implementasi dan pengujian pada optimasi model MobileNetV2 untuk pendekripsi Bahasa Isyarat Indonesia (BISINDO) dengan analisis dan perencanaan yang telah dibuat.

4.1 Implementasi

Pada bagian ini dilakukan implementasi terdiri dari beberapa tahap, yaitu pengembangan model *deep learning* MobileNetV2 dan penerapan sistem *real-time*.

4.1.1 Implementasi Model Deep Learning

Penelitian ini menggunakan dataset gestur BISINDO yang telah dikembangkan sebelumnya. Dataset ini mencakup 9 kosakata utama dalam BISINDO yaitu “A”, “Asal”, “B”, “C”, “D”, “E”, “Halo”, “I”, “Terima Kasih”, dengan masing-masing kosakata terdiri dari 1.000 sampel gambar, sehingga total dataset yang digunakan dalam penelitian ini berjumlah 9.000 sampel gambar. Dataset ini telah melalui proses pengumpulan dan penyusunan dengan mempertimbangkan berbagai kondisi seperti variasi sudut pandang, pencahayaan, serta perbedaan individu yang melakukan gestur, guna meningkatkan keberagaman dan generalisasi model.

Sebelum digunakan dalam proses *training* model, dataset yang telah tersedia di Kaggle harus diunduh dan diekstrak ke dalam Google Colab menggunakan kode pada Gambar 4.1. Proses ini diawali dengan menginstal dan mengonfigurasi API Kaggle. Langkah pertama adalah menginstal *library* Kaggle dan mengunggah file kaggle.json, yang berisi API key untuk autentikasi ke akun Kaggle. Setelah file berhasil diunggah, dataset dapat diunduh. Setelah dataset berhasil diunduh, file yang berbentuk arsip ZIP kemudian diekstrak ke dalam direktori.

```
!pip install -q Kaggle
from google.colab import files
files.upload()
!kaggle datasets download -d tessaagitha/bisindo-by-elcue
!mkdir bisindo-by-elcue
!unzip bisindo-by-elcue.zip -d bisindo-by-elcue
!ls bisindo-by-elcue
```

Gambar 4.1 Code untuk Memuat Dataset BISINDO dari Kaggle

Setelah dataset berhasil dimuat, tahap *preprocessing* data dilakukan untuk meningkatkan kualitas citra sebelum digunakan dalam pelatihan model dengan menggunakan kode pada Gambar 4.2. Proses ini dimulai dengan memuat dan mengelompokkan gambar berdasarkan kelasnya, di mana setiap gambar dikonversi ke *grayscale*, diproses menggunakan teknik *edge detection*, dan diubah ukurannya menjadi 64x64 piksel agar sesuai dengan arsitektur model. Setelah itu, nilai piksel dinormalisasi dalam skala 0-1 untuk mempercepat konvergensi saat pelatihan.

Dataset kemudian dibagi menjadi *training set* (80%) dan *testing set* (20%) guna memastikan evaluasi model dilakukan secara objektif. Selanjutnya, data dilengkapi dengan augmentasi menggunakan *ImageDataGenerator*, yang menerapkan *rotation*, *translation*, *zooming*, dan *flipping horizontal*. Untuk mencegah bias dalam urutan data, seluruh data pelatihan dan pengujian diacak menggunakan metode *shuffle* sebelum digunakan dalam tahap selanjutnya.

```

from sklearn.utils import shuffle
from tensorflow.keras.preprocessing.image import ImageDataGenerator

data_folder = '/content/bisindo-by-elcue/dataset_isyarat'

def load_images():
    images = []
    labels = []
    index = -1
    folders = sorted(os.listdir(data_folder))
    for folder in folders:
        index += 1
        print("Loading images from folder ", folder , " has started.")
        for image in os.listdir(data_folder + '/' + folder):
            img = cv2.imread(data_folder + '/' + folder + '/' + image, 0)
            img = edge_detection(img)
            img = cv2.resize(img, (64, 64))
            img = img_to_array(img)
            images.append(img)
            labels.append(index)
    images = np.array(images)
    images = images.astype('float32')/255.0
    labels = to_categorical(labels)
    x_train, x_test, y_train, y_test = train_test_split(images, labels,
test_size=0.2)
    return x_train, x_test, y_train, y_test

def edge_detection(image):
    minValue = 70
    blur = cv2.GaussianBlur(image,(5,5),2)
    th3 =
cv2.adaptiveThreshold(blur,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BI
NARY_INV,11,2)
    ret, res = cv2.threshold(th3, minValue, 255,
cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
    return res

x_train, x_test, y_train, y_test = load_images()
x_train, y_train = shuffle(x_train, y_train, random_state=17)
x_test, y_test = shuffle(x_test, y_test, random_state=17)

datagen = ImageDataGenerator(
    rotation_range=15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
datagen.fit(x_train)

```

Gambar 4.2 Code untuk Tahap *Preprocessing* Data

Setelah tahap *preprocessing data*, model *deep learning* dibangun menggunakan arsitektur MobileNetV2, yang telah di-*training* sebelumnya (*pre-trained model*) pada dataset ImageNet. Untuk menyesuaikan dengan dataset BISINDO, model dikonfigurasi ulang dengan mengubah ukuran input menjadi (64, 64, 3) serta menambahkan *additional layers* untuk klasifikasi. Lapisan awal MobileNetV2 dibekukan (*trainable=False*) agar bobot yang telah dipelajari tetap dipertahankan, sementara *additional layers* ditambahkan berupa lapisan *Global Average Pooling*, *Fully Connected Layer* dengan 128 neuron beraktivasi ReLU, dan *Dropout* sebesar 30% untuk mencegah *overfitting*. Lapisan terakhir menggunakan Softmax dengan 9 output neuron untuk mengklasifikasikan 9 kelas gestur BISINDO.

Model di-*compile* menggunakan *optimizer* Adam dengan *learning rate* 0.001, serta *loss function* *sparse categorical crossentropy*. Setelah proses inisialisasi, dilakukan *fine-tuning* dengan membuka sebagian lapisan MobileNetV2 setelah layer ke-100 untuk memungkinkan model beradaptasi dengan karakteristik dataset BISINDO. Dalam tahap *fine-tuning*, *learning rate* diturunkan menjadi 0.00001 agar pembaruan bobot lebih stabil dan model tidak kehilangan fitur yang telah dipelajari dari dataset sebelumnya. Model yang telah selesai dibangun kemudian ditampilkan menggunakan *model.summary()* untuk memastikan struktur arsitekturnya sesuai. Adapun kode untuk pembuatan model ini terdapat pada Gambar 4.3.

```

map_characters = {0: 'A', 1: 'Asal', 2: 'B', 3: 'C', 4: 'D', 5: 'E', 6: 'Halo', 7: 'I', 8: 'Terima kasih'}

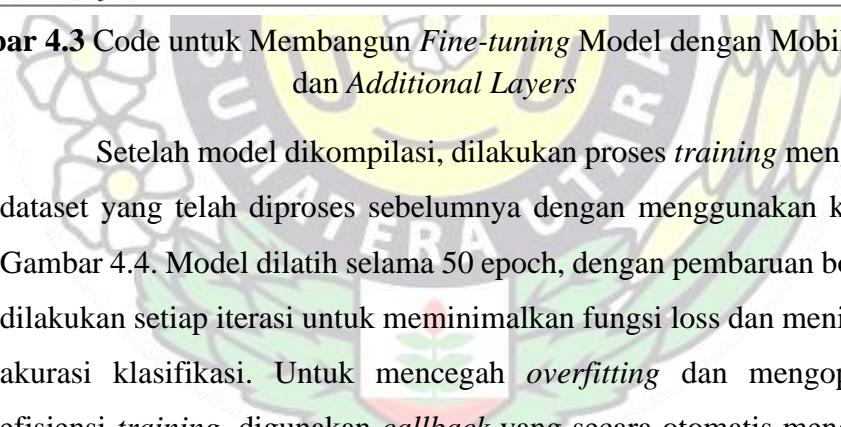
x_train = np.repeat(x_train, 3, axis=-1)
x_test = np.repeat(x_test, 3, axis=-1)
base_model = MobileNetV2(input_shape=(64, 64, 3), include_top=False,
weights='imagenet')
base_model.trainable = False

model = models.Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(9, activation='softmax')
])
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

base_model.trainable = True
for layer in base_model.layers[:100]:
    layer.trainable = False
model.compile(optimizer=Adam(learning_rate=0.00001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.summary()

```

Gambar 4.3 Code untuk Membangun *Fine-tuning* Model dengan MobileNetV2 dan *Additional Layers*



Setelah model dikompilasi, dilakukan proses *training* menggunakan dataset yang telah diproses sebelumnya dengan menggunakan kode pada Gambar 4.4. Model dilatih selama 50 epoch, dengan pembaruan bobot yang dilakukan setiap iterasi untuk meminimalkan fungsi loss dan meningkatkan akurasi klasifikasi. Untuk mencegah *overfitting* dan mengoptimalkan efisiensi *training*, digunakan *callback* yang secara otomatis menghentikan pelatihan jika akurasi model telah mencapai lebih dari 99%. Model dilatih dengan menggunakan data pelatihan (*x_train*, *y_train*) dan diuji pada data pengujian (*x_test*, *y_test*). Proses pelatihan dilakukan dengan pengacakan data (*shuffle=True*) untuk memastikan model tidak dipengaruhi oleh urutan data. Selama proses pelatihan, metrik evaluasi seperti akurasi dan *loss* dicatat untuk dianalisis lebih lanjut pada tahap evaluasi model.

```

class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if logs.get('accuracy') is not None and (logs.get('accuracy') >
0.99):
            print("\nCancelling training")
            self.model.stop_training = True

classes = 9
epochs = 50
callback = myCallback()
history = model.fit(x_train, y_train,
                     epochs=epochs,
                     verbose=2,
                     validation_data=(x_test, y_test),
                     shuffle=True,
                     callbacks=[callback])

```

Gambar 4.4 Code untuk *Training Model*

4.1.2 Implementasi Model secara Real-time

Setelah model *deep learning* berhasil dilatih, model disimpan dalam format TensorFlow Lite (TFLite) dengan menggunakan kode pada Gambar 4.5. TFLite dipilih karena memiliki ukuran model yang lebih ringan dan dioptimalkan untuk inferensi *real-time*, terutama dalam aplikasi yang berjalan di perangkat *mobile* atau *embedded system*. Proses konversi model ke format TFLite dilakukan dengan menggunakan TensorFlow Lite Converter, yang mengubah model dari format TensorFlow SavedModel atau H5 menjadi format yang lebih ringan dan dapat dijalankan tanpa dependensi penuh dari TensorFlow.

```

model = tf.keras.models.load_model('model_elcue.h5')
converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
tflite_model = converter.convert()
with open('model_elcue.tflite', 'wb') as f:
    f.write(tflite_model)

```

Gambar 4.5 Code untuk konversi model ke TFLite

Setelah konversi selesai, model kemudian digunakan untuk inferensi real-time menggunakan OpenCV, yang bertugas menangkap citra dari kamera, melakukan *preprocessing* seperti konversi *grayscale*, *resizing*, dan normalisasi, serta menjalankan model TFLite Interpreter untuk melakukan prediksi gestur BISINDO secara langsung. Adapun implementasi kodennya terdapat pada Gambar 4.6.

```

interpreter = tflite.Interpreter(model_path="model_elcue.tflite")
interpreter.allocate_tensors()
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
input_shape = input_details[0]['shape']
img_size = (input_shape[1], input_shape[2])
map_characters = {0: 'A', 1: 'Asal', 2: 'B', 3: 'C', 4: 'D', 5: 'E', 6:
'Halo', 7: 'I', 8: 'Terima kasih'}

def preprocess_frame(frame):
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    frame = cv2.resize(frame, img_size)
    frame = np.expand_dims(frame, axis=-1)
    frame = np.repeat(frame, 3, axis=-1)
    frame = np.expand_dims(frame, axis=0)
    frame = frame.astype(np.float32) / 255.0

cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret:
        break
    input_data = preprocess_frame(frame)
    interpreter.set_tensor(input_details[0]['index'], input_data)
    interpreter.invoke()
    output_data = interpreter.get_tensor(output_details[0]['index'])
    predicted_class = np.argmax(output_data)
    predicted_label = map_characters.get(predicted_class, "Tidak
Diketahui")
    cv2.putText(frame, f"Prediksi: {predicted_label}", (20, 50),
               cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2, cv2.LINE_AA)
    cv2.imshow("Real-time BISINDO Detection", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()

```

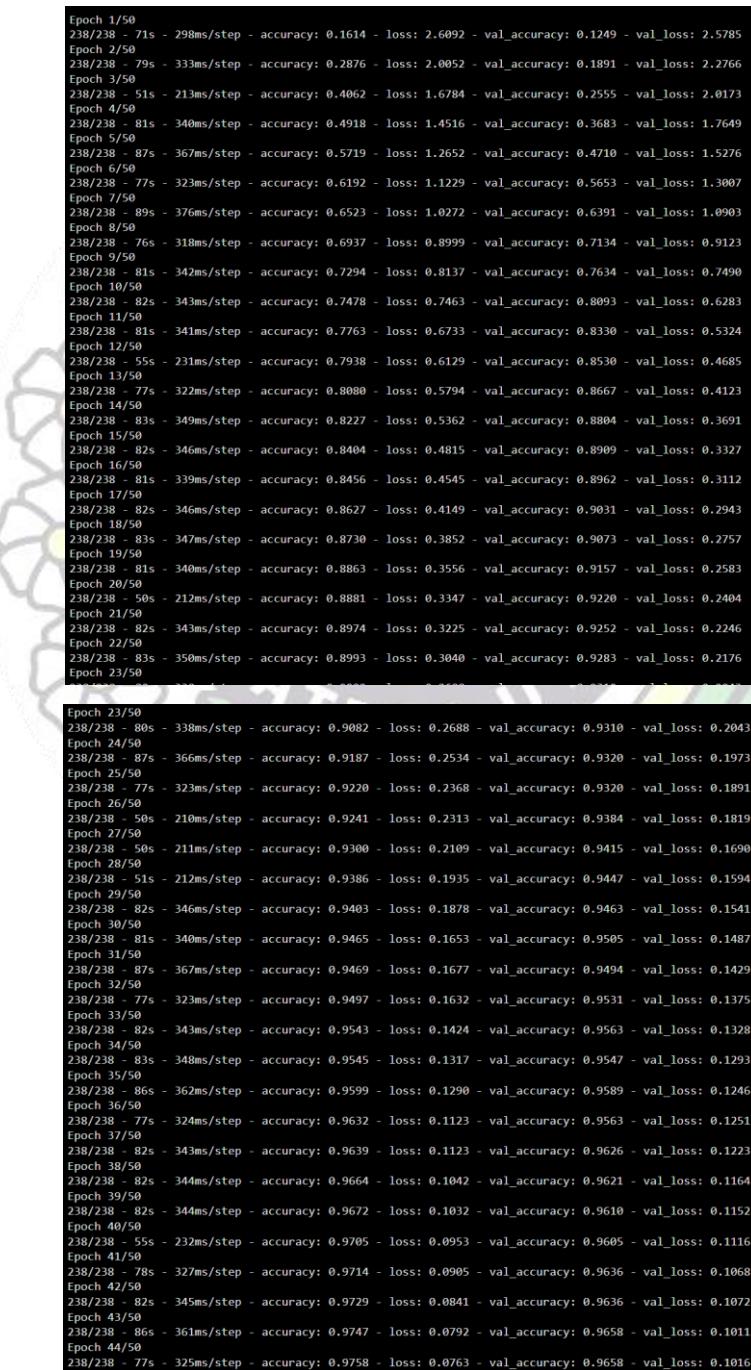
Gambar 4.6 Code untuk implementasi model secara *real-time*

4.2 Pengujian

Pengujian dilakukan untuk mengevaluasi akurasi dan performa model dalam mendeteksi Bahasa Isyarat Indonesia (BISINDO) secara *real-time*.

4.2.1 Pengujian Akurasi Model

Performa model selama *training* dapat dilihat melalui Gambar 4.7, yang menampilkan perkembangan nilai loss dan akurasi pada setiap epoch selama proses *training*.



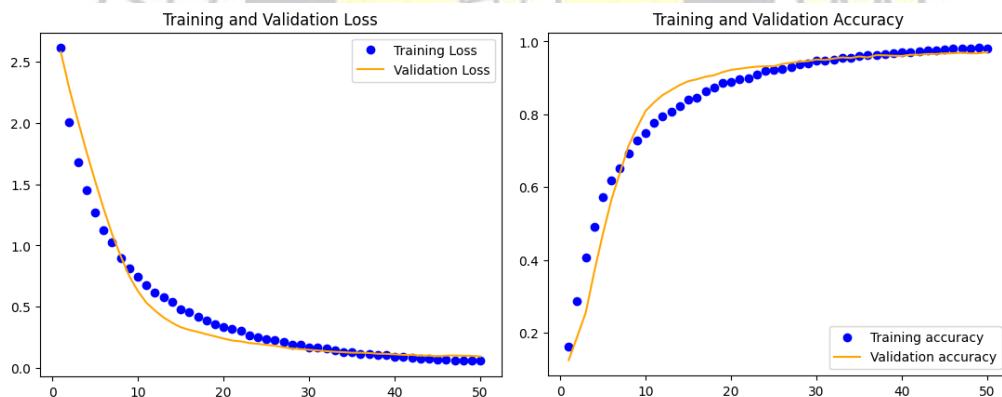
```

Epoch 45/50
238/238 - 81s - 342ms/step - accuracy: 0.9788 - loss: 0.0706 - val_accuracy: 0.9673 - val_loss: 0.0954
Epoch 46/50
238/238 - 50s - 212ms/step - accuracy: 0.9798 - loss: 0.0667 - val_accuracy: 0.9679 - val_loss: 0.1012
Epoch 47/50
238/238 - 83s - 347ms/step - accuracy: 0.9800 - loss: 0.0648 - val_accuracy: 0.9694 - val_loss: 0.1019
Epoch 48/50
238/238 - 51s - 215ms/step - accuracy: 0.9818 - loss: 0.0610 - val_accuracy: 0.9673 - val_loss: 0.1001
Epoch 49/50
238/238 - 51s - 216ms/step - accuracy: 0.9821 - loss: 0.0575 - val_accuracy: 0.9679 - val_loss: 0.0987
Epoch 50/50
238/238 - 81s - 341ms/step - accuracy: 0.9806 - loss: 0.0585 - val_accuracy: 0.9710 - val_loss: 0.0944

```

Gambar 4.7 Performa selama proses training dengan 50 epoch

Grafik *training* dan *validation loss* pada Gambar 4.8 menunjukkan bahwa nilai *loss* model mengalami penurunan yang konsisten seiring bertambahnya epoch, baik untuk data latih maupun data validasi. Pada awal pelatihan, *loss* relatif tinggi, namun setelah sekitar 10 epoch, *loss* mulai menurun secara signifikan dan mendekati nol pada epoch ke-50. Grafik *training* dan *validation accuracy* pada Gambar 4.x menunjukkan bahwa akurasi model meningkat secara signifikan pada epoch awal, lalu mengalami stabilisasi mendekati nilai maksimum (~97-99%) setelah sekitar 20 epoch. Tidak terdapat kesenjangan besar antara akurasi training dan validasi, yang mengindikasikan bahwa model tidak mengalami *overfitting*, sehingga dapat menggeneralisasi dengan baik pada data baru.



Gambar 4.8 Grafik *training* and *validation* untuk *loss* dan *accuracy*

Berdasarkan hasil pengujian pada Gambar 4.9, model mencapai akurasi sebesar 97.1%, menunjukkan bahwa model memiliki tingkat keberhasilan tinggi dalam mengenali gestur BISINDO. Nilai *precision*, *recall*, dan F1-score untuk masing-masing gestur berada dalam rentang 0.92 hingga 0.99, yang menunjukkan keseimbangan antara kemampuan model dalam mengidentifikasi gestur yang benar (*precision*) dan mendeteksi seluruh gestur yang ada dalam dataset (*recall*).

Pada bagian akhir evaluasi, terdapat dua nilai rata-rata, yaitu *macro average* dan *weighted average*, yang masing-masing menunjukkan nilai rata-rata metrik evaluasi tanpa memperhitungkan jumlah sampel per kelas (macro avg: 0.97) serta nilai rata-rata yang mempertimbangkan jumlah sampel di setiap kelas (weighted avg: 0.97). Ini menunjukkan bahwa model memiliki performa yang stabil di seluruh kelas dan tidak menunjukkan bias signifikan terhadap kelas tertentu. Secara keseluruhan, hasil ini menunjukkan bahwa model yang telah di-*training* memiliki akurasi tinggi dalam mengenali gestur BISINDO.

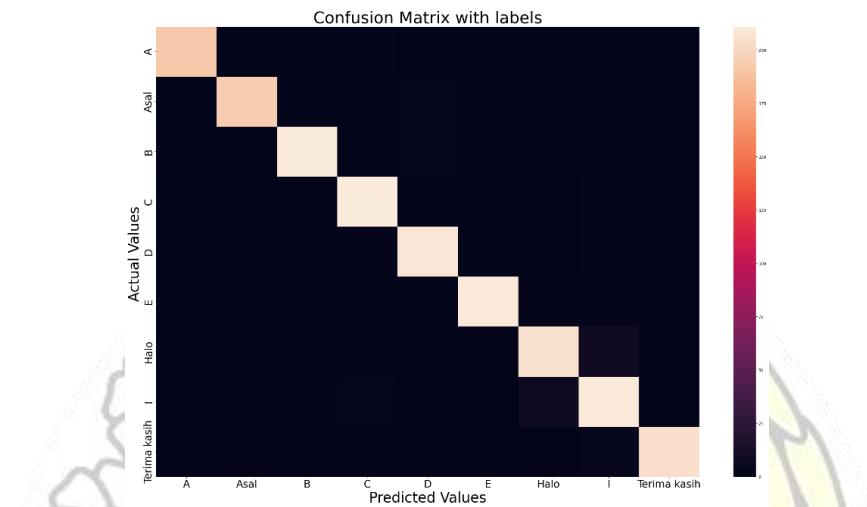
[0.09444785863161087, 0.9710221290588379]				
Keras CNN - accuracy: 0.9710221290588379				
60/60 ━━━━━━━━ 10s 141ms/step				
	precision	recall	f1-score	support
A	0.99	0.98	0.99	195
Asal	0.98	0.98	0.98	198
B	0.99	0.98	0.99	215
C	0.96	0.97	0.97	217
D	0.97	0.97	0.97	214
E	0.99	0.98	0.98	214
Halo	0.95	0.94	0.95	217
I	0.92	0.95	0.94	219
Terima kasih	0.99	0.98	0.98	209
accuracy			0.97	1898
macro avg	0.97	0.97	0.97	1898
weighted avg	0.97	0.97	0.97	1898

Gambar 4.9 Evaluasi Model

Pada *confusion matrix* Gambar 4.10 dan Gambar 4.11, sumbu vertikal (*Actual Values*) menunjukkan kelas sebenarnya, sedangkan sumbu horizontal (*Predicted Values*) menunjukkan kelas yang diprediksi oleh model. Dari hasil *confusion matrix*, terlihat bahwa sebagian besar nilai berada di sepanjang diagonal utama, yang menunjukkan bahwa model memiliki tingkat prediksi yang sangat baik, dengan hanya sedikit kesalahan klasifikasi. Sebagai contoh, untuk kelas A, model mengklasifikasikan 192 sampel dengan benar, tetapi terdapat 2 sampel yang salah diklasifikasikan sebagai C, 1 sebagai E, dan 1 sebagai kelas lainnya.

Beberapa kesalahan klasifikasi yang muncul cenderung terjadi pada kelas dengan pola gestur yang mungkin mirip atau memiliki fitur yang

tumpang tindih, seperti pada kelas Halo dan I, di mana beberapa sampel kelas I salah diprediksi sebagai Halo. Namun, jumlah kesalahan ini relatif kecil dibandingkan dengan jumlah total sampel, sehingga dapat disimpulkan bahwa model memiliki tingkat akurasi tinggi dan kesalahan klasifikasi yang minim.



Gambar 4.10 Confusion Matrix

Confusion Matrix									
[[192	0	0	2	1	0	0	0	0	0]
[0 194	1	0	3	0	0	0	0	0	0]
[1 0 211	0	3	0	0	0	0	0	0	0]
[0 1 1 211	0	1	1	1	2	0	0	0	0]
[0 1 0 2 208	0	0	2	1	0	0	2	1	0]
[0 0 0 1 0 209	2	0	0	0	209	2	2	0	0]
[0 0 0 1 0 1 205	9	1	0	0	0	1	205	9	1]
[0 0 0 3 0 0 7 209	0	0	0	0	0	7	209	0	0]
[0 1 0 0 0 0 1 3 204]]									

Gambar 4.11 Confusion Matrix

Tabel 4.1 menyajikan perbandingan penelitian ini dengan beberapa penelitian terkait, termasuk metode yang digunakan, dataset yang dipakai, akurasi yang diperoleh,

Tabel 4.1 Perbandingan Hasil Evaluasi Model dengan Penelitian Sebelumnya

Peneliti	Model	Dataset	Akurasi
Shania dkk. (2022)	Xception	11 Kosakata BISINDO	98.5%
Wijaya dkk. (2024)	Artificial Neural Network (ANN)	SIBI (Alfabet & Kata)	96.15% (Alfabet) / 99.45% (Kata)

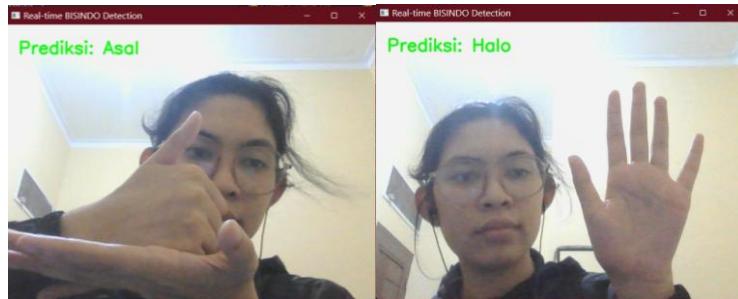
Aldhahri dkk. (2022)	MobileNet	Arabic Alphabets Sign Language Dataset (ArASL2018)	94.46%
Penelitian ElCue (2024)	MobileNetV2 + Additional Layers	9 Kosakata BISINDO (250 gambar per kelas)	94%
Penelitian ElCue (2025)	MobileNetV2 + Fine-tuning + Additional Layers	9 Kosakata BISINDO (1,000 gambar per kelas)	97.1%

4.2.2 Pengujian Performa secara Real-time

Pengujian performa secara *real-time* dilakukan untuk mengevaluasi kemampuan model dalam mengenali gestur BISINDO secara langsung menggunakan kamera laptop (CPU). Model diuji dengan menampilkan prediksi secara *real-time*, di mana setiap kelas gestur dilakukan sebanyak 20 kali untuk mengukur seberapa akurat model dalam mendeteksi gerakan tangan dalam kondisi dunia nyata. Hasil pengujian dicatat dengan menghitung jumlah prediksi yang benar dan salah untuk masing-masing kelas, kemudian dihitung tingkat akurasi *real-time* berdasarkan persentase prediksi yang benar terhadap jumlah total percobaan. Hasil pengujian performa secara *real-time* dapat dilihat pada Tabel 4.2.

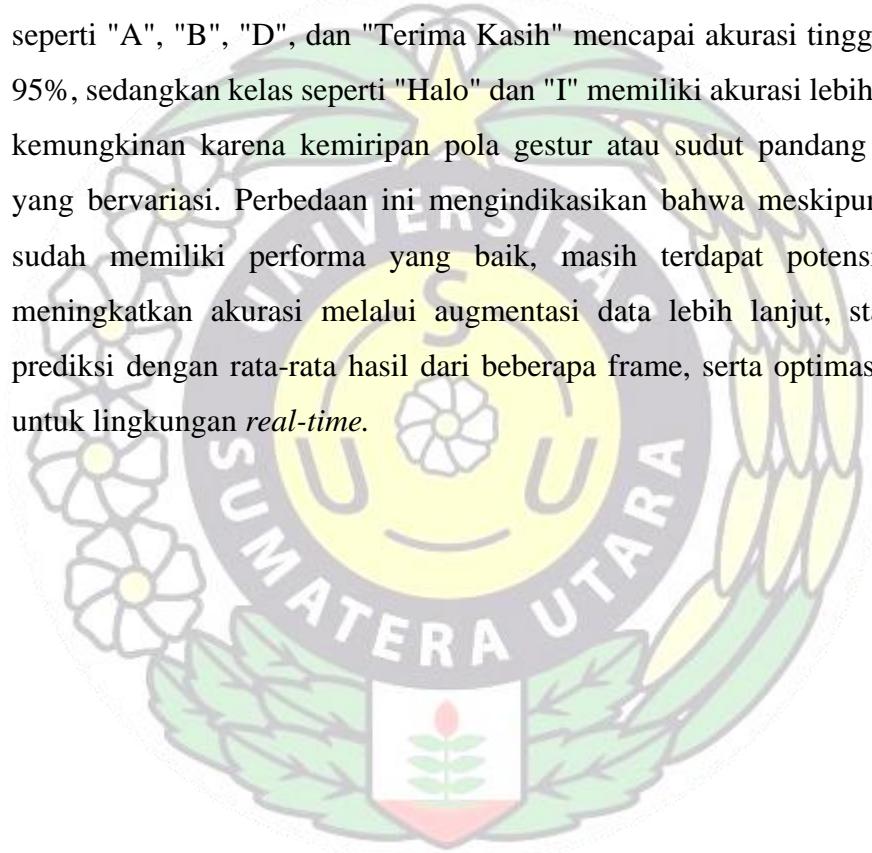
Tabel 4.2 Performa Model ketika Inferensi Real-time

Kelas Gestur	Jumlah Percobaan	Prediksi	Prediksi	Akurasi (%)
		Benar	Salah	
A	20	19	1	95.0%
Asal	20	18	2	90.0%
B	20	19	1	95.0%
C	20	17	3	85.0%
D	20	19	1	95.0%
E	20	18	2	90.0%
Halo	20	16	4	80.0%
I	20	15	5	75.0%
Terima kasih	20	19	1	95.0%
Rata-rata	180	160	20	88.9%



Gambar 4.12 Inferensi Model secara Real-time

Hasil pengujian menunjukkan bahwa model memiliki akurasi rata-rata sebesar 88.9% dalam kondisi *real-time*, dengan beberapa kelas gestur seperti "A", "B", "D", dan "Terima Kasih" mencapai akurasi tinggi di atas 95%, sedangkan kelas seperti "Halo" dan "I" memiliki akurasi lebih rendah, kemungkinan karena kemiripan pola gestur atau sudut pandang kamera yang bervariasi. Perbedaan ini mengindikasikan bahwa meskipun model sudah memiliki performa yang baik, masih terdapat potensi untuk meningkatkan akurasi melalui augmentasi data lebih lanjut, stabilisasi prediksi dengan rata-rata hasil dari beberapa frame, serta optimasi model untuk lingkungan *real-time*.



BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah tahap implementasi dilakukan dalam penelitian optimasi model MobileNetV2 dengan *fine-tuning* dan *additional layers* untuk pendekripsi Bahasa Isyarat Indonesia (BISINDO) secara *real-time* pada aplikasi *video call* ElCue, serta melalui pengujian terhadap akurasi model dan performa *real-time*, maka diperoleh kesimpulan sebagai berikut:

1. MobileNetV2 terbukti dapat digunakan sebagai *feature extractor* dalam sistem pendekripsi BISINDO, dengan proses *transfer learning* dan *fine-tuning*, yang memungkinkan model untuk mengenali gestur tangan secara efektif.
2. Teknik *fine-tuning* pada MobileNetV2 terbukti meningkatkan akurasi model, di mana model mencapai akurasi 97.1% pada dataset uji dan 88.9% dalam kondisi *real-time*, yang menunjukkan performa yang baik dalam mendekripsi gestur BISINDO.
3. Penambahan *additional layers* setelah fitur diekstraksi dari MobileNetV2, seperti *Global Average Pooling*, *Dense Layer*, dan *Dropout*, membantu meningkatkan stabilitas model dalam melakukan klasifikasi gestur secara lebih akurat.
4. Pengujian performa *real-time* menunjukkan bahwa akurasi model dapat menurun pada kondisi pencahayaan tertentu atau sudut kamera yang berbeda, yang mengindikasikan perlunya augmentasi data lebih lanjut dan optimasi preprocessing agar model lebih adaptif terhadap kondisi lingkungan yang bervariasi.
5. MobileNetV2 memiliki keunggulan dalam efisiensi komputasi dibandingkan model lain yang lebih kompleks, namun tetap memiliki tantangan dalam menangani gestur dengan pola yang mirip.

5.2 Saran

Berdasarkan hasil penelitian ini, terdapat beberapa aspek yang dapat dikembangkan lebih lanjut untuk meningkatkan akurasi, efisiensi, dan kestabilan model dalam

pendekstrian BISINDO secara *real-time*. Beberapa saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Untuk meningkatkan ketahanan model terhadap variasi pencahayaan, sudut pandang, dan latar belakang, penelitian selanjutnya dapat menerapkan augmentasi data yang lebih luas, termasuk teknik seperti *adaptive brightness adjustment*, *motion blur simulation*, dan *random occlusion* untuk mensimulasikan kondisi dunia nyata secara lebih menyeluruh.
2. Untuk meningkatkan akurasi dalam pengenalan gestur yang kompleks, penelitian selanjutnya dapat menggabungkan model CNN dengan *pose estimation* seperti MediaPipe.
3. Model yang dikembangkan dapat lebih dioptimalkan untuk inferensi real-time pada perangkat *mobile* atau *embedded system* dengan menggunakan teknik *quantization* (INT8) atau *pruning* untuk mengurangi ukuran model tanpa mengorbankan akurasi secara signifikan.

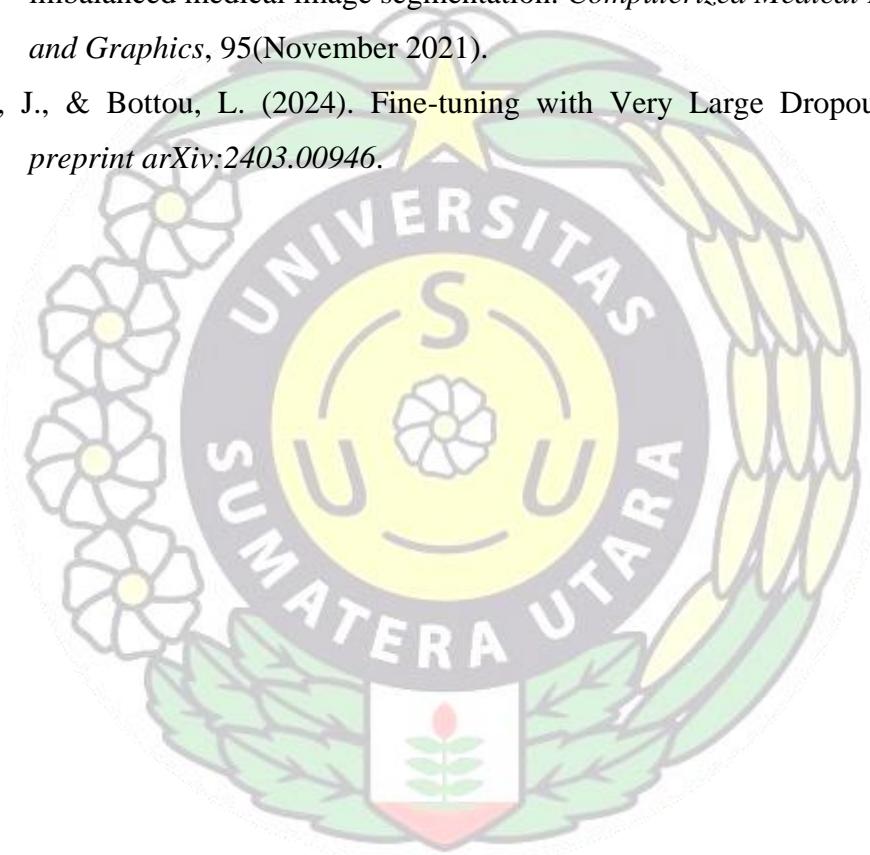


DAFTAR PUSTAKA

- Aldhahri, E., Aljuhani, R., Alfaidi, A., Alshehri, B., Alwadei, H., Aljojo, N., Alshutayri, A. & Almazroi, A. (2023). Correction to: Arabic Sign Language Recognition Using Convolutional Neural Network and MobileNet. *Arabian Journal for Science and Engineering*, 48(2), 2147-2154.
- Arjaria, S. K., Rathore, A. S., & Cherian, J. S. (2021). Kidney disease prediction using a machine learning approach: A comparative and comprehensive analysis. *Demystifying Big Data, Machine Learning, and Deep Learning for Healthcare Analytics*, 307–333.
- Fitroh, Q. A. (2023). Deep Transfer Learning untuk Meningkatkan Akurasi Klasifikasi pada Citra Dermoskopi Kanker Kulit. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 12(2), 78-84.
- Hikmatia, N. A. E., dan Zul, M. I. (2021). Aplikasi Penerjemah Bahasa Isyarat Indonesia menjadi Suara berbasis Android menggunakan Tensorflow. *Jurnal Politeknik Caltex Riau*. 7(1):74-83.
- Howard, A. G., Wang, W., Zhu, M., Weyand, T., Chen, B., Andreetto, M., Kalenichenko, D., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- IBM. (2023). *What is regularization?* IBM. Diakses melalui <https://www.ibm.com/think/topics/regularization>
- Kalayci, T. A., & Asan, U. (2022). Improving classification performance of fully connected layers by fuzzy clustering in transformed feature space. *Symmetry*, 14(4), 658.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- LeCun, Y., Bengio, Y., dan Hinton, G. (2015). Deep learning. *nature*, 521(7553): 436-444.

- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Li, X., Herranz, L., & Jiang, S. (2020). Multifaceted analysis of fine-tuning in a deep model for visual recognition. *ACM Transactions on Data Science*, 1(1), 1-22.
- Liew, X. Y., Hameed, N., & Clos, J. (2021). An investigation of XGBoost-based 60 algorithm for breast cancer classification. *Machine Learning with Applications*, 6(April), 100154.
- Peryanto, A., Yudhana, A., & Umar, R. (2020). Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation. *Journal of Applied Informatics and Computing*, 4(1), 45–51.
- Pu, H., & Yi, K. (2024). A Comparative Analysis of EfficientNet and MobileNet Models' Performance on Limited Datasets: An Example of American Sign Language Alphabet Detection. *Highlights in Science, Engineering and Technology*, 94, 558-564.
- Rizky, A. D. B., Faqihuddin, M. A., Romadhan, F. F., dan Siradjuddin, I. A. 2023. Identifikasi Alfabet Bahasa Isyarat Indonesia dengan Menggunakan Convolutional LSTM. *Seminar Nasional Sinergitas Era Digital 5.0 dalam Pembangunan Teknologi Hijau Berkelanjutan* (pp. 183-190).
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4510-4520).
- Shania, S., Naufal, M. F., Prasetyo, V. R., & Azmi, M. S. B. (2022). Translator of Indonesian Sign Language Video using Convolutional Neural Network with Transfer Learning. *Indonesian Journal of Information Systems*, 5(1), 17-27.
- Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)* (pp. 464-472). IEEE.

- Temkar, R., Jagtap, S., Jadhav, K., & Deshmukh, M. (2024). Real-time Sign Language Recognition Using MobileNetV2 and Transfer Learning. *arXiv preprint arXiv:2412.07486*.
- Wijaya, F., Dahendra, L., Purwanto, E. S., & Ario, M. K. (2024). Quantitative analysis of sign language translation using artificial neural network model. *Procedia Computer Science*, 245, 998-1009.
- Yeung, M., Sala, E., Schönlieb, C. B., & Rundo, L. (2022). Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95(November 2021).
- Zhang, J., & Bottou, L. (2024). Fine-tuning with Very Large Dropout. *arXiv preprint arXiv:2403.00946*.



Lampiran : 3 (tiga) lembar

Hal : Permohonan Persetujuan Dosen Wali Akademik
untuk Bentuk Lain Setara Skripsi

Yth. Bapak Dr. Mohammad Andri Budiman S.T., M.Comp.Sc., M.E.M.

Dosen Pembimbing Akademik

Program Studi S-1 Ilmu Komputer

Fakultas Ilmu Komputer dan Teknologi Informasi

Universitas Sumatera Utara

Medan

Dengan hormat, berdasarkan Keputusan Kepala Balai Pengembangan Talenta Indonesia, Pusat Prestasi Nasional, Sekretariat Jenderal Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Nomor 1653/J7.1/PN.00/2024 tentang Penetapan Pemenang Pekan Ilmiah Mahasiswa Nasional (PIMNAS) Ke-37 Tahun 2024 dimana saya dan tim lolos sebagai pemenang Medali Emas kategori Poster bidang PKM-Karsa Cipta. Oleh karena itu saya memohon untuk dapat mengajukannya sebagai Bentuk Lain Setara Skripsi untuk penyelesaian Tugas Akhir saya. Adapun judul yang akan saya ajukan sebagai berikut:

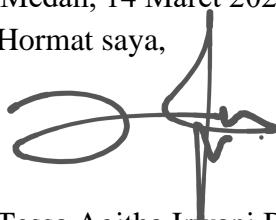
”Optimasi MobileNetV2 dengan *Fine-tuning* dan *Additional Layers* untuk Pendekripsi Bahasa Isyarat Indonesia (BISINDO) secara *Real-time* dalam Aplikasi Video Call ElCue”

Bersama surat ini saya lampirkan dokumen pendukung untuk pengajuan Bentuk lain Setara Skripsi.

Demikian permohonan ini disampaikan, atas perhatian Bapak diucapkan terima kasih.

Medan, 14 Maret 2025

Hormat saya,



Tessa Agitha Irvani Br Barus
NIM. 211401138

**SURAT PERSETUJUAN DOSEN PEMBIMBING AKADEMIK
TENTANG BENTUK LAIN SETARA SKRIPSI**

Saya yang bertanda tangan di bawah ini:

Nama : Dr. Mohammad Andri Budiman S.T., M.Comp.Sc., M.E.M.
NIP : 197510082008011011
Pangkat/Golongan : Lektor/IIIId
Program Studi : S-1 Ilmu Komputer
Fakultas : Ilmu Komputer dan Teknologi Informasi - USU

Menyetujui permohonan untuk mengajukan Bentuk Lain Setara Skripsi untuk memenuhi Tugas Akhir sebagai syarat kelulusan mahasiswa sebagai berikut:

Nama : Tessa Agitha Irwani Br Barus
NIM : 211401138
Program Studi : S-1 Ilmu Komputer
Bentuk Lain Setara Skripsi : Karya Ilmiah yang dimenangkan pada Pekan Ilmiah Nasional (PIMNAS) ke-37 Tahun 2024 (Medali Emas)
Judul : Optimasi MobileNetV2 dengan *Fine-tuning* dan *Additional Layers* untuk Pendekripsi Bahasa Isyarat Indonesia (BISINDO) secara *Real-time* dalam Aplikasi *Video Call ElCue*
Demikian disampaikan agar dapat dipergunakan dengan sebaiknya.

Medan, 14 Maret 2025

Dosen Pembimbing Akademik,


Dr. Mohammad Andri Budiman
S.T., M.Comp.Sc., M.E.M.
NIP 197510082008011011



SURAT KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
NOMOR: 1530/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 : Bahwa 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/UNS.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 :

"Optimasi MobileNetV2 dengan Fine-tuning dan Additional Layers untuk Pendekripsi Bahasa Isyarat Indonesia (BISINDO) secara Real-time dalam Aplikasi Video Call ElCue"

Memutuskan

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 : Tessa Agitha Irwani Br Barus
NIM : 211401138
Program Studi : S-1 Ilmu Komputer Fasilkom-TI USU
2. Dosen Pembimbing:
1. Hayatunnufus, S.Kom., M.Cs. (Pembimbing Pertama)
NIP 199207192024062001
2. Dr. M. Andri Budiman, S.T., M.Comp.Sc., M.E.M. (Pembimbing Kedua)
NIP 197510082008011011
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 ternyata 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: Penguatan 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

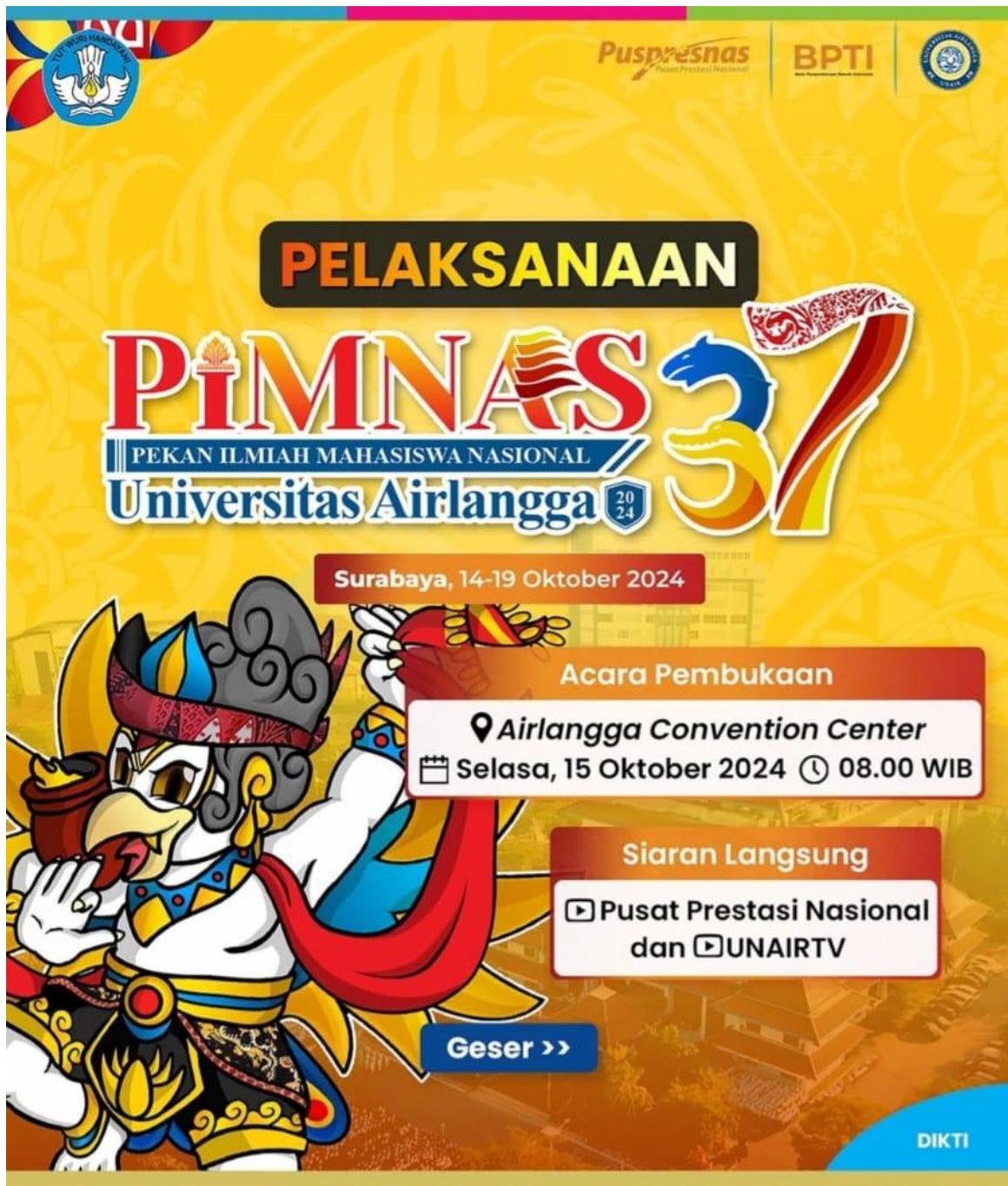
DOKUMENTASI PENGUMUMAN PIMNAS KE-37



SERTIFIKAT PESERTA DAN PEMENANG PIMNAS KE-37



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 fine-tuning pada model MobileNetV2. Mendesain dan menguji tambahan layer (custom layers) untuk meningkatkan akurasi deteksi. Mengukur performa model (akurasi, latency) pada input real-time.
2	Putri Andriyani / 211401008	S-1 Ilmu Komputer	<ul style="list-style-type: none"> Membangun aplikasi ElCue dengan Flutter. Mengintegrasikan layanan backend (Firebase) dan video call (Agora). Menyambungkan model deteksi BISINDO ke tampilan aplikasi secara real-time.
3	Rani Widya Astuti / 211401018	S-1 Ilmu Komputer	<ul style="list-style-type: none"> Meneliti dan menerapkan teknik pre-processing berupa augmentasi data dan edge detection. Memastikan model tetap robust terhadap variasi gerakan tangan dalam kondisi real-time. Bekerja sama membuat model untuk memastikan hasil pre-processing 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 labeling, anotasi data, dan validasi kualitas data. Mengembangkan pipeline inferensi berbasis Computer Vision 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 deep learning dan speech recognition ke sistem ElCue. Mengatur alur komunikasi antar input gesture dan suara untuk meningkatkan komunikasi 2 arah. Menjamin sinkronisasi dan performa dari dua jalur input (gesture + voice).



PRIMARY SOURCES

1	text-id.123dok.com Internet Source	2%
2	Submitted to Universitas Sumatera Utara Student Paper	2%
3	repository.unja.ac.id Internet Source	2%
4	repositori.usu.ac.id Internet Source	1 %
5	Aldin Fathiray, Joni Maulindar, Wiji Lestari. "Pengembangan Sistem Penerjemah Kalimat Bahasa Isyarat Bisindo To Text Dengan Kinect Real Time", Infotek: Jurnal Informatika dan Teknologi, 2025 Publication	1 %
6	Submitted to Universitas Brawijaya Student Paper	<1 %
7	dspace.uii.ac.id Internet Source	<1 %
8	Ananda Rizki Dani, Irma Handayani. "Classification of Yogyakarta Batik Motifs Using GLCM and CNN Methods", Jurnal Teknologi Terpadu, 2024 Publication	<1 %
9	www.etonline-digitallibrary.com Internet Source	<1 %