

**ANALISIS PERBANDINGAN PERFORMA ALGORITMA VISION
TRANSFORMER, RESNET, DAN EFFICIENTNETV2 DALAM
KLASIFIKASI ALFABET BAHASA ISYARAT INDONESIA (BISINDO)**

A FUAD AHSAN BASIR

H071201076



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR**

2025

**ANALISIS PERBANDINGAN PERFORMA ALGORITMA VISION
TRANSFORMER, RESNET, DAN EFFICIENTNETV2 DALAM
KLASIFIKASI ALFABET BAHASA ISYARAT INDONESIA (BISINDO)**

A FUAD AHSAN BASIR

H071201076

Skripsi

sebagai salah satu syarat untuk mencapai gelar sarjana

Program Studi Sistem Informasi

pada

**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2025**

SKRIPSI

ANALISIS PERBANDINGAN PERFORMA ALGORITMA VISION TRANSFORMER, RESNET, DAN EFFICIENTNETV2 DALAM KLASIFIKASI ALFABET BAHASA ISYARAT INDONESIA (BISINDO)

A FUAD AHSAN BASIR
H071201076

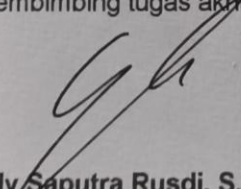
Skripsi,

telah dipertahankan di depan Panitia Ujian Sarjana Sistem Informasi pada 21 Maret
2025 dan dinyatakan telah memenuhi syarat kelulusan
pada

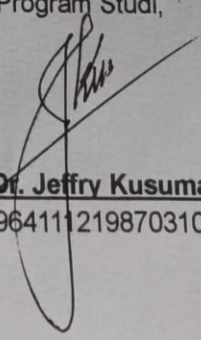
Program Studi Sistem Informasi
Departemen Matematika
Fakultas Matematika & Ilmu Pengetahuan Alam
Universitas Hasanuddin
Makassar



Mengesahkan:
Pembimbing tugas akhir,


Edy Saputra Rusdi, S.Si., M.Si.
NIP. 199104102020053001

Mengetahui:
Ketua Program Studi,


Prof. Dr. Jeffry Kusuma, Ph. D.
NIP. 196411121987031002

PERNYATAAN KEASLIAN SKRIPSI DAN PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa, skripsi berjudul "Analisis Perbandingan Performa Algoritma Vision Transformer, Resnet, dan Efficientnetv2 dalam Klasifikasi Alfabet Bahasa Isyarat Indonesia (BISINDO)" adalah benar karya saya dengan arahan dari pembimbing Edy Saputra Rusdi, S.Si., M.Si. sebagai Pembimbing Utama. Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka skripsi ini. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini adalah karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut berdasarkan aturan yang berlaku.

Dengan ini saya melimpahkan hak cipta (hak ekonomis) dari karya tulis saya berupa skripsi ini kepada Universitas Hasanuddin.

Makassar, 21 Maret 2025



A Fuad Ahsan Basir
H071201076

UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan atas kehadiran Allah SWT. atas berkat rahmat dan hidayah-Nya, penulis dapat menyelesaikan tugas akhir “Analisis Perbandingan Performa Algoritma Vision Transformer, Resnet, dan Efficientnetv2 dalam Klasifikasi Alfabet Bahasa Isyarat Indonesia (BISINDO).” Penulis menyadari bahwa tugas akhir ini dapat terselesaikan dengan baik berkat bantuan dan dukungan dari berbagai pihak.

Pertama-tama, penulis ingin memberikan ucapan terima kasih yang sebesar-besarnya kepada Ayah dan Ibu penulis yang telah mendidik penulis dengan penuh kesabaran dan penuh kasih sayang serta mendukung perjalanan hidup penulis selama ini. Penulis juga mengucapkan terima kasih atas bantuan saudara-saudara penulis atas do'a serta dukungan yang diberikan selama ini. Penulis juga mengucapkan terima kasih kepada keluarga penulis yang tidak dapat disebutkan satu-satu karena telah menjadi pondasi pendukung penulis.

Penulis mengucapkan terima kasih kepada bapak Edy Saputra Rusdi, S.Si., M.Si. yang telah menyempatkan waktunya dalam membimbing penulis serta memberikan pengetahuan dan kebijaksanaannya sehingga tugas akhir ini dapat terselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada bapak Prof. Dr. Eng. Armin Lawi, S.Si., M.Eng. dan bapak Jeriko Gormantara, S.Si., M.Si. atas seluruh saran dan masukan yang diberikan sehingga dapat menutupi kekurangan yang muncul selama penyusunan tugas akhir ini. Tak lupa pula penulis mengucapkan terima kasih kepada seluruh staf dan dosen universitas, fakultas, departemen, dan program studi atas bantuan dan dukungannya selama penulis berkuliah di kampus Universitas Hasanuddin ini.

Penulis juga menyampaikan ucapan terima kasih kepada teman-teman Sistem Informasi 2020 yang telah menemani penulis serta membantu penulis baik di lingkup perkuliahan maupun di luar lingkup perkuliahan. Penulis juga memberikan ucapan terima kasih kepada rekan-rekan asisten Laboratorium Sistem Informasi atas kerjasamanya selama kelas praktikum sehingga dapat terlaksanakan dengan baik. Penulis juga mengucapkan terima kasih kepada rekan tim dan mentor-mentor yang membimbing penulis selama menjalani Bangkit 2023 sehingga penulis dapat mendapatkan benefit tidak hanya pengetahuan teknikal tetapi juga keterampilan dasar yang sangat bermanfaat untuk kedepannya. Terima kasih juga saya ucapkan kepada rekan-rekan KKNT gelombang 111 atas dedikasi dan kesabarannya dalam menanggapi berbagai macam pelak-pelik masalah yang muncul selama menjalankan pengabdian masyarakat.

Penulis,
A. Fuad Ahsan Basir

ABSTRAK

A. FUAD AHSAN BASIR. **Analisis Perbandingan Performa Algoritma Vision Transformer, Resnet, dan Efficientnetv2 dalam Klasifikasi Alfabet Bahasa Isyarat Indonesia (BISINDO)** (dibimbing oleh Edy Saputra Rusdi, S.Si., M.Si.).

Sebagian individu memanfaatkan bahasa isyarat sebagai sarana komunikasi karena keterbatasan fisik yang mereka miliki namun tidak semua orang mampu memahami bahasa isyarat dengan baik. Dalam konteks ini, teknologi klasifikasi gambar telah menjadi solusi yang signifikan untuk mengatasi berbagai masalah di berbagai bidang. Penelitian ini bertujuan untuk mendapatkan hasil kinerja algoritma *EfficientNetV2-S*, *ResNet-50*, dan *ViT-B/32* dalam mengklasifikasikan huruf alfabet BISINDO. Dataset yang digunakan adalah citra huruf alfabet BISINDO yang didapatkan dari menggabungkan data yang diperoleh oleh peneliti dan data yang dikumpulkan dari halaman *Kaggle* dan *Roboflow* dengan total keseluruhan sebanyak 2600 citra. Penelitian ini menggunakan akurasi, *loss*, dan *F1-score* untuk mengukur kinerja setiap model. Model yang memiliki kinerja terbaik akan diuji kembali menggunakan input secara *real-time*. Model *ViT-B/32* memiliki kinerja terbaik dalam semua metrik pengujian jika dibandingkan dengan *EfficientNetV2-S* dan *ResNet-50*. Akurasi *ViT-B/32* memiliki akurasi *testing* tertinggi yaitu sebesar 91,92%, diikuti oleh *ResNet-50* dengan akurasi sebesar 89,23% dan *EfficientNetV2-S* sebesar 81,92%. Model *ViT-B/32* juga memiliki stabilitas paling baik ditandai dengan nilai *loss* terendah yaitu 0,29 diikuti oleh *ResNet-50* dengan nilai *loss* sebesar 0,47 dan *EfficientNetV2-S* dengan nilai *loss* sebesar 0,67. Melihat dari rata-rata *F1-score*, *ViT-B/32* memiliki rata-rata *F1-score* sebesar 0,92 diikuti oleh *ResNet-50* dengan *F1-score* 0,89 dan *EfficientNetV2-S* dengan *F1-score* sebesar 0,82. Selain itu, model *ViT-B/32* dapat memprediksi secara tepat 89 dari 100 frame gambar dalam pengujian *real-time*. Model *ViT-B/32* tidak hanya memiliki kinerja yang paling baik dibandingkan dengan dua model lainnya, tetapi juga mampu mempertahankan kinerja yang baik ketika diuji saat mengklasifikasikan huruf BISINDO secara *real-time*.

Kata kunci: BISINDO; EfficientNetV2; perbandingan algoritma; Residual Network (ResNet); Vision Transformer (ViT).

ABSTRACT

A. FUAD AHSAN BASIR. **Comparative Performance Analysis of Vision Transformer, ResNet, and EfficientNetV2 Algorithms in Classifying Indonesian Sign Language (BISINDO) Alphabet** (Supervised by Edy Saputra Rusdi, S.Si., M.Si).

Some individuals use sign language as a means of communication due to their physical limitations; however, not everyone is able to understand sign language well. In this context, image classification technology has become a significant solution for addressing various problems across different fields. This research aims to evaluate the performance of the EfficientNetV2-S, ResNet-50, and ViT-B/32 algorithms in classifying BISINDO alphabet letters. The dataset used consists of BISINDO alphabet letter images obtained by combining data collected by researchers with data gathered from Kaggle and Roboflow, totaling 2,600 images. This study uses accuracy, loss, and F1-score to measure the performance of each model. The best-performing model will be further tested using real-time input. Among the three models, ViT-B/32 demonstrated the best performance across all evaluation metrics. It achieved the highest testing accuracy of 91.92%, followed by ResNet-50 with 89.23%, and EfficientNetV2-S with 81.92%. ViT-B/32 also exhibited the best stability, with the lowest loss value of 0.29, compared to ResNet-50's 0.47 and EfficientNetV2-S's 0.67. In terms of the average F1-score, ViT-B/32 scored 0.92, followed by ResNet-50 at 0.89 and EfficientNetV2-S at 0.82. Additionally, ViT-B/32 was able to accurately predict 89 out of 100 image frames in real-time testing. This indicates that ViT-B/32 not only outperforms the other two models in overall performance but also maintains its effectiveness when classifying BISINDO letters in real-time.

Keywords: BISINDO; EfficientNetV2; algorithm comparison; Residual Network (ResNet); Vision Transformer (ViT).

DAFTAR ISI

UCAPAN TERIMA KASIH	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Landasan Teori.....	4
1.6.1 Bahasa Isyarat Indonesia (BISINDO)	4
1.6.2 Neural Networks	4
1.6.3 Convolutional Neural Network	5
1.6.4 Transfer Learning.....	8
1.6.5 Residual Network.....	8
1.6.6 Vision Transformer.....	10
1.6.7 Efficient Network	11
1.6.8 Confusion Matrix.....	12
BAB II METODE PENELITIAN.....	14
2.1 Waktu dan Tempat Penelitian	14
2.1.1 Waktu Penelitian	14
2.1.2 Tempat Penelitian	14
2.2 Tahap Penelitian	14
2.2.1 Pengumpulan Dataset	15
2.2.2 Preprocessing.....	16
2.2.3 Split Data	16
2.2.4 Augmentasi Citra	16
2.2.5 Data Loader dan Hyperparameter	16
2.2.6 Transfer Learning dan Fine-Tuning Model	17
2.2.7 Evaluasi	17

2.3 Instrumen Penelitian.....	17
BAB III HASIL DAN PEMBAHASAN	18
3.1 Deskripsi Data	18
3.2 Preprocessing	18
3.3 Split Data.....	18
3.4 Augmentasi Citra	18
3.5 Hyperparameter	19
3.6 Transfer Learning	19
3.6.1 EfficientNetV2-S.....	20
3.6.2 ResNet-50.....	21
3.6.3 ViT-B/32	22
3.7 Evaluasi dengan Dataset Primer	23
3.7.1 Akurasi dan Loss	23
3.7.2 Confusion Matrix.....	24
3.7.3 Real-Time Testing	31
3.8 Evaluasi dengan Merged Dataset	34
3.8.1 Akurasi dan Loss	34
3.8.2 Confusion Matrix.....	35
3.8.3 Real-Time Testing	42
BAB IV PENUTUP	45
4.1 Kesimpulan.....	45
4.2 Saran.....	45
DAFTAR PUSTAKA	46
LAMPIRAN	49

DAFTAR TABEL

Tabel 1. Confusion Matrix	13
Tabel 2. Waktu penelitian	14
Tabel 3. Sumber Data.....	16
Tabel 4. Setelan Hyperparameter	19
Tabel 5. Arsitektur EfficientNetV2-S.....	20
Tabel 6. Arsitektur ResNet-50.....	21
Tabel 7. Arsitektur ViT-B/32	22
Tabel 8. Precision, recall, dan F1-score dari model EfficientNetV2-S dengan data primer sebagai data pelatihan.....	26
Tabel 9. Precision, recall, dan F1-score dari model ResNet-50 dengan Data Primer sebagai Data Pelatihan.....	28
Tabel 10. Precision, recall, dan F1-score dari model ViT-B/32 dengan Data Primer sebagai Data Pelatihan.....	30
Tabel 11. Hasil Pengujian Klasifikasi Huruf Vokal dengan ViT-B/32 yang dilatih dengan data primer.....	31
Tabel 12. Precision, recall, dan F1-score dari model EfficientNetV2-S dengan Merged Dataset sebagai data pelatihan.....	37
Tabel 13. Precision, recall, dan F1-score dari model ResNet-50 dengan Merged Dataset sebagai data pelatihan.....	39
Tabel 14. Precision, recall, dan F1-score dari model ViT-B/32 dengan Merged Dataset sebagai data pelatihan.....	41
Tabel 15. Hasil Pengujian Klasifikasi Huruf Vokal dengan ViT-B/32 yang dilatih dengan Merged Dataset	42

DAFTAR GAMBAR

Gambar 1. Alfabet dalam BISINDO (Buku Saku Bahasa Isyarat, Kementerian Kesehatan)	4
Gambar 2. Contoh Neural Network	5
Gambar 3. Proses perhitungan operasi konvolusi.....	6
Gambar 4. Average Pooling	7
Gambar 5. Max Pooling.....	7
Gambar 6. Grafik Fungsi Aktivasi ReLU.....	7
Gambar 7. Jaringan residual (Bishop dan Bishop, 2024).	9
Gambar 8. Bottleneck residual	9
Gambar 9. Alur diagram modul attention dan multi-attention (Peng & Wang, 2022).	10
Gambar 10. Struktur MBConv dan Fused-MBConv	12
Gambar 11. Alur Diagram Penelitian	15
Gambar 12. Citra Asli (Kiri) dan Citra Augmentasi (Kanan).....	19
Gambar 13. Kurva Akurasi Model dengan Data Primer sebagai Data Pelatihan....	23
Gambar 14. Kurva Loss Model dengan Data Primer sebagai Data Pelatihan.....	24
Gambar 15. Confusion Matrix EfficientNetV2-S dengan Data Primer sebagai Data Pelatihan.....	25
Gambar 16. Confusion Matrix ResNet-50 dengan Data Primer sebagai Data Pelatihan.....	27
Gambar 17. Confusion Matrix ViT-B/32 dengan Data Primer sebagai Data Pelatihan	29
Gambar 18. Kurva akurasi model dengan Merged Dataset sebagai data pelatihan	34
Gambar 19. Kurva Loss model dengan Merged Dataset sebagai data pelatihan ..	35
Gambar 20. Confusion Matrix EfficientNetV2-S dengan Merged Dataset sebagai data pelatihan	36
Gambar 21. Confusion Matrix ResNet-50 dengan Merged Dataset sebagai data pelatihan	38
Gambar 22. Confusion Matrix ViT-B/32 dengan Merged Dataset sebagai data pelatihan	40

BAB I

PENDAHULUAN

1.1 Latar Belakang

Manusia sebagai makhluk sosial memerlukan interaksi, komunikasi, dan sosialisasi dengan manusia lainnya. Komunikasi memegang peran yang krusial dalam berinteraksi dan berhubungan dengan orang lain. Seseorang dapat mengekspresikan berbagai pikirannya kepada orang lain sehingga mencapai pemahaman bersama atas pesan tersebut melalui komunikasi. Komunikasi dilakukan secara verbal maupun non-verbal. Komunikasi verbal melibatkan penggunaan kata-kata untuk berinteraksi, sedangkan komunikasi non-verbal menggunakan gerakan atau bahasa tubuh untuk menyampaikan pesan (Gustiar et al., 2020)

Secara umum, bahasa yang digunakan untuk berkomunikasi adalah bahasa verbal. Akan tetapi, pada beberapa kasus tertentu komunikasi secara verbal tidak dapat dilakukan dikarenakan keterbatasan fisik seperti tunarungu dan tunawicara. Penderita tunarungu dan tunawicara menggunakan bahasa isyarat sebagai cara untuk berkomunikasi dengan sekitarnya. Bahasa isyarat menggunakan bahasa tubuh dengan melibatkan gerakan tangan, mimik wajah, dan gerakan tubuh untuk merepresentasikan sebuah huruf atau kata (Nur'azizan et al., 2024). Berdasarkan data Badan Pusat Statistik mengenai Jumlah Penduduk Berumur 5 Tahun ke Atas menurut Kelompok Umur, Daerah Perkotaan/Perdesaan, Jenis Kelamin, dan Tingkat Kesulitan Memahami atau Dipahami Orang Lain ketika Berbicara di Indonesia Tahun 2022 bahwa terdapat 3.234.964 orang mengalami kesulitan berkomunikasi dengan rincian 310.056 orang sama sekali tidak dapat berkomunikasi / berbicara / memahami / dipahami, 574.153 orang memiliki banyak kesulitan, dan 2.350.755 orang mengalami sedikit kesulitan.

Terdapat dua bahasa isyarat yang digunakan di Indonesia yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). SIBI merupakan bahasa isyarat Indonesia yang dikembangkan dari American Sign Language (ASL) dan BISINDO merupakan bahasa awal yang digunakan oleh tunarungu yang berkembang serta dipakai dalam kehidupan sehari-hari (Arisandi dan Satya, 2022). BISINDO diprakarsai oleh Gerakan Kesejahteraan Tunarungu Indonesia (GERKATIN) kemudian dikembangkan oleh komunitas tunarungu (Arifah et al., 2022) dan menjadi bahasa yang sering digunakan oleh komunitas tunarungu karena mudah dipahami (Dewi et al., 2021). Meskipun BISINDO mudah dipahami, masih ada tantangan dalam berkomunikasi antara pengguna bahasa isyarat dan orang yang berbicara dengan bahasa verbal yang belum pernah mempelajari bahasa isyarat sebelumnya. Salah satu metode yang dapat dilakukan untuk mengatasi tantangan komunikasi tersebut adalah dengan klasifikasi gerakan tangan bahasa isyarat dengan cara menerapkan model *Machine Learning*.

Salah satu metode pendekatan yang biasa digunakan dalam tugas klasifikasi gambar adalah *Convolutional Neural Network* (CNN). Salah satu arsitektur CNN

yang populer digunakan adalah *Residual Network* (ResNet) (Shabbir et al., 2021). Metode lain yang dapat digunakan untuk klasifikasi gambar adalah *Vision Transformer* (ViT). *Transformer* memiliki berbagai keunggulan seperti model depedensi jarak jauh, adaptasi dengan berbagai ukuran *input* serta potensi untuk mempercepat proses pelatihan dengan pemrosesan paralel. Meskipun begitu, CNN juga memiliki keunggulan dalam efisiensi komputasi, kebutuhan data latihan yang lebih sedikit, dan robust terhadap noise yang lebih kecil (Maurício et al., 2023).

Penelitian yang dilakukan oleh Reedha et al. (2021) dengan judul "*Vision Transformers For Weeds and Crops Classification Of High Resolution UAV Images*" untuk mendeteksi/mengenali pesawat nirawak (drone) menghasilkan akurasi yang lebih tinggi pada model *Vision Transformer* (ViT) dibandingkan dengan model *ResNet-50*, *EfficientNet-B0*, dan *EfficientNet-B1*. Dalam penelitian lain yang dilakukan oleh Alharthi dan Alzahrani (2023) dengan judul "*Vision Transformers and Transfer Learning Approaches for Arabic Sign Language Recognition*" untuk mengukur performa model *Vision Transformer* (ViT) dalam melakukan klasifikasi bahasa isyarat arab. Penelitian ini menghasilkan akurasi untuk model *ResNet-50* yang lebih tinggi daripada model ViT.

Seiring dengan perkembangan penelitian, berbagai model telah dikembangkan lebih lanjut untuk mendapatkan hasil yang lebih baik dari *Vision Transformer* dan *Residual Network*. Penelitian yang dilakukan oleh Tan dan Le, (2021) menghasilkan sebuah model yang memiliki akurasi lebih tinggi dari *ViT-B/32* dan *ResNet-50* terhadap dataset *ImageNet* dan *CIFAR* yaitu *EfficientNetV2*. Penelitian ini juga menyebutkan bahwa kecepatan *training* dari *EfficientNetV2* lebih cepat daripada model-model lainnya.

Berdasarkan penelitian-penelitian sebelumnya, kinerja dari setiap algoritma memiliki performa yang berbeda bergantung pada masalah yang dihadapi. Maka dari itu, diperlukan pengujian dan perbandingan lebih lanjut mengenai kinerja algoritma dalam melakukan klasifikasi gambar. Penerapan algoritma *Vision Transformer Base 32* (ViT-B/32), *Residual Network 50* (ResNet-50), dan *Efficient Network V2 Small* (EfficientNetV2-s) dalam klasifikasi bahasa isyarat Indonesia sendiri masih belum banyak dilakukan. Oleh karena itu, peneliti tertarik melakukan penelitian untuk membandingkan kinerja algoritma *ViT-B/32*, *ResNet-50*, dan *EfficientNetV2* pada klasifikasi huruf alfabet Bahasa Isyarat Indonesia (BISINDO).

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang, terdapat beberapa masalah yang dapat dirumuskan menjadi:

1. Bagaimana membangun model *ViT-B/32*, *ResNet-50*, *EfficientNetV2-S* klasifikasi huruf alfabet Bahasa Isyarat Indonesia (BISINDO) menggunakan *transfer learning*?
2. Bagaimana perbandingan kinerja algoritma *ViT-B/32*, *ResNet-50*, *EfficientNetV2-S* pada klasifikasi huruf alfabet Bahasa Isyarat Indonesia (BISINDO) ketika dilatih menggunakan data primer?

3. Bagaimana perbandingan kinerja algoritma *ViT-B/32*, *ResNet-50*, *EfficientNetV2-S* pada klasifikasi huruf alfabet Bahasa Isyarat Indonesia (BISINDO) ketika dilatih menggunakan data gabungan?
4. Bagaimana performa model dengan kinerja terbaik ketika diuji secara *real-time* menggunakan webcam?

1.3 Batasan Masalah

Batasan masalah yang diterapkan pada penelitian ini adalah:

1. Dataset yang digunakan adalah gambar huruf alfabet Bahasa Isyarat Indonesia (BISINDO) yang terbatas pada 26 kelas (A-Z).
2. Data yang digunakan adalah data primer yang dibuat oleh peneliti dan data sekunder yang diambil dengan menggabungkan beberapa dataset yang terdapat dari halaman *kaggle* dan *roboflow*.
3. Arsitektur yang digunakan adalah *ViT-B/32*, *ResNet-50*, dan *EfficientNetV2-S* dengan bobot *pre-trained*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Membangun model *ViT-B/32*, *ResNet-50*, *EfficientNetV2-S* untuk mengklasifikasikan huruf alfabet Bahasa Isyarat Indonesia (BISINDO) menggunakan *transfer learning*.
2. Menganalisis kinerja yang dihasilkan dari model *ViT-B/32*, *ResNet-50*, dan *EfficientNetV2-S* dalam mengklasifikasi alfabet Bahasa Isyarat Indonesia (BISINDO) ketika dilatih menggunakan data primer.
3. Menganalisis kinerja yang dihasilkan dari model *ViT-B/32*, *ResNet-50*, dan *EfficientNetV2-S* dalam mengklasifikasi alfabet Bahasa Isyarat Indonesia (BISINDO) ketika dilatih menggunakan data gabungan.
4. Memvalidasi kemampuan model yang memiliki kinerja tertinggi untuk diuji lebih lanjut pada pengujian *real-time* menggunakan webcam.

1.5 Manfaat Penelitian

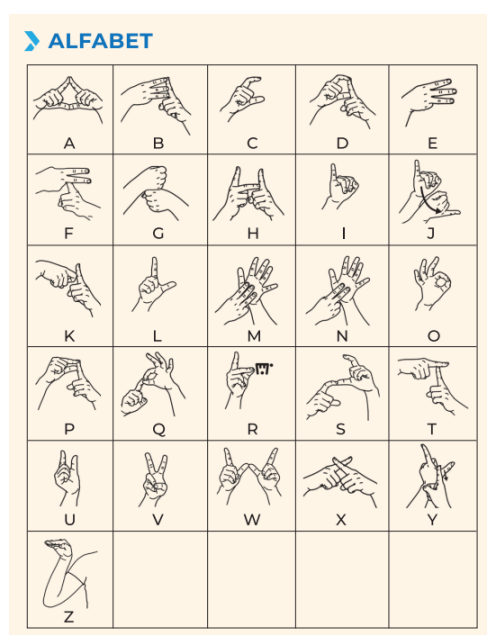
Manfaat dari penelitian ini adalah:

1. Mendapatkan informasi lebih dalam mengenai penerapan *ViT-B/32*, *ResNet-50*, dan *EfficientNetV2-S* dalam mengklasifikasi alfabet Bahasa Isyarat Indonesia (BISINDO).
2. Menghasilkan model *Machine Learning* (*ViT-B/32*, *ResNet-50*, dan *EfficientNetV2-S*) yang dapat digunakan dalam mengklasifikasi alfabet Bahasa Isyarat Indonesia (BISINDO).
3. Mendapatkan informasi kinerja model terbaik ketika diuji secara *real-time*.

1.6 Landasan Teori

1.6.1 Bahasa Isyarat Indonesia (BISINDO)

BISINDO merupakan salah satu media komunikasi yang digunakan oleh komunitas tunarungu di Indonesia. Teman Tuli lebih terbiasa menggunakan BISINDO dikarenakan penyusunan dan pengembangannya disesuaikan dengan penggunaan bahasa dalam kehidupan sehari-hari (Fadillah et al., 2021). BISINDO diprakarsai oleh Gerakan Kesejahteraan Tunarungu Indonesia (GERKATIN) kemudian dikembangkan oleh komunitas tunarungu (Arifah et al., 2022) dan menjadi bahasa yang sering digunakan oleh komunitas tunarungu karena mudah dipahami (Dewi et al., 2021). BISINDO menggunakan kedua tangan untuk mengekspresikan huruf alfabet.



Gambar 1. Alfabet dalam BISINDO (Buku Saku Bahasa Isyarat, Kementerian Kesehatan)

1.6.2 Neural Networks

Neural Networks merupakan model komputasi yang pada mulanya terinspirasi dari cara kerja otak manusia dalam belajar dan mengelola informasi dan menjadi dasar dari *Deep Learning*. Unit dasar yang melakukan pemrosesan di otak dikenal sebagai *Neuron*. Karakteristik dari *neuron* direpresentasikan dalam model matematika yang sederhana, yang disebut sebagai jaringan syaraf tiruan (*Artificial Neural Networks*). *Neural Networks* meniru karakteristik sel tersebut kemudian menjadi dasar dari

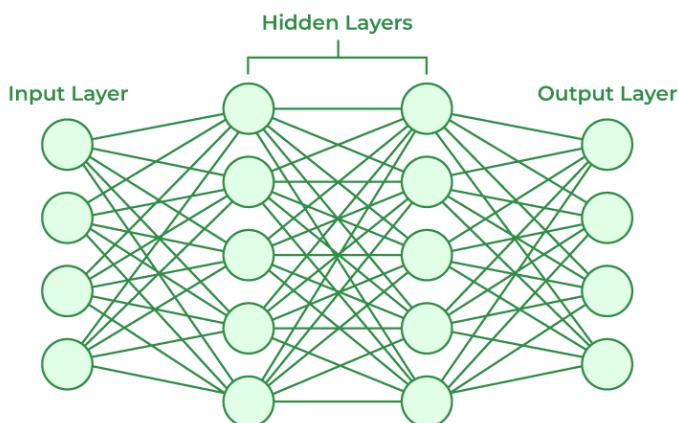
pendekatan komputasional dalam pembelajaran. Ekspresi matematikanya dapat dituliskan dalam:

$$a = \sum_{i=1}^M w_i x_i \quad (1.1)$$

$$y = f(a) \quad (1.2)$$

Dimana $x_i, \dots, x_M$ mewakili M *inputs neuron* dan $w_i, \dots, w_M$ merepresentasikan bobot (*weight*) yang menghubungkan input ke *neuron*. Nilai a dikenal sebagai pra-aktivasi, fungsi f disebut fungsi aktivasi, dan output y disebut aktivasi (Bishop dan Bishop, 2024).

Dalam *Neural Networks*, *neuron* berperan untuk menerima input, melakukan operasi matematika, dan mengirimkan output ke neuron lain atau memberikan hasil akhir. Satu *Neuron* tidak cukup untuk menyelesaikan masalah pembelajaran yang rumit, seperti membedakan angka pada tulisan tangan. Maka dari itu, kumpulan *neuron* diatur untuk bekerja secara bersama-sama dalam kumpulan *layer*.



Gambar 2. Contoh *Neural Network*

Lapisan di sebelah kiri bertugas untuk menerima data masukan yang disebut sebagai *Input Layer*. Lapisan di sebelah kanan bertugas untuk memberikan hasil akhir yang disebut sebagai *Output Layer*. Lapisan di tengah bertugas untuk melakukan pemrosesan data serta pembelajaran representasi fitur dan pola dalam data yang disebut sebagai *Hidden Layer* (Buduma & Locascio, 2017).

1.6.3 Convolutional Neural Network

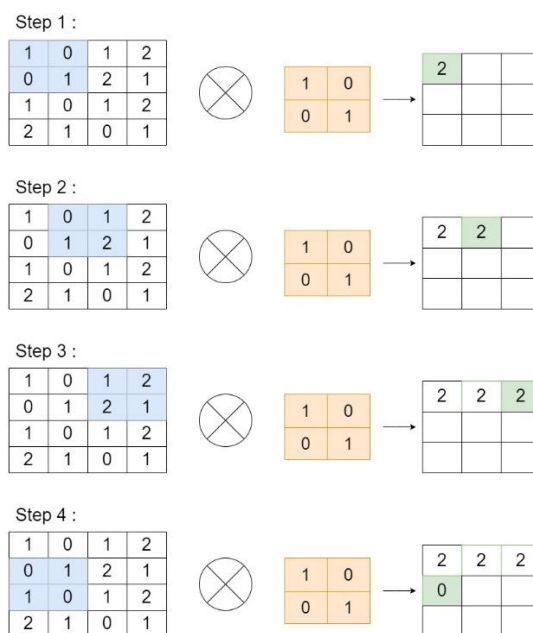
Convolutional Neural Network (CNN) merupakan algoritma *deep learning* yang mampu untuk menerima gambar sebagai input, menentukan pentingnya berbagai aspek atau objek dalam gambar melalui *weight* dan *bias*, serta membedakan satu objek dengan objek yang lain. Kebutuhan *preprocessing* CNN cenderung lebih rendah daripada algoritma klasifikasi lainnya. Terdapat tiga jenis *layer* pada CNN, yaitu *input layer*, *hidden layer*, dan *output layer*. *Hidden layer* sendiri terdiri atas

Convolution layer, fungsi aktivasi ReLU (*Rectifier Linear Unit*), *pooling layer*, *normalization layer* serta *fully connected layer* (Elngar et al., 2021). Berikut ini penjelasan secara detail mengenai tahap arsitektur CNN.

Convolution Layer. Lapisan konvolusi memiliki peran yang penting dalam arsitektur CNN. Tujuan utama dari lapisan konvolusi adalah mendeteksi motif lokal yang khas seperti tepi, garis, dan elemen visual lainnya. Operasi matematika dalam lapisan ini melibatkan perkalian tetangga lokal dari suatu piksel dengan array kecil dari parameter yang disebut *kernel* (Lee et al., 2017). Persamaan operasi konvolusi pada citra I dan *kernel* filter K dituliskan pada persamaan (1.3).

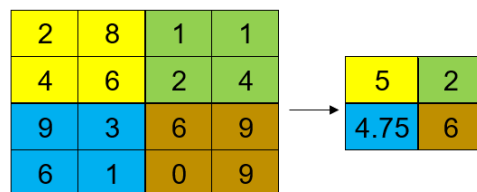
$$(I * K)(i, j) = \sum_{m=0}^{k_1} \sum_{n=0}^{k_2} I(i + m, j + n) K(m, n) \quad (1.3)$$

Dimana $I(i, j)$ mewakili *tensor* citra input dengan koordinat posisi i, j dan $K(k_1, k_2)$ merupakan filter *kernel* dengan ukuran $k_1 \times k_2$. Proses konvolusi dilakukan dengan cara menempatkan filter *Kernel* ke atas input dan menghitung *dot product*. Selanjutnya, kernel digeser secara horizontal dan vertikal ke seluruh area gambar input, dan dot product dihitung kembali pada setiap posisi pergeseran. Proses ini akan terus dilakukan sampai kernel tidak dapat digeser lagi (Alzubaidi et al., 2021). *Kernel* memiliki ukuran yang lebih kecil dari inputan (Teuwen dan Moriakov, 2020). Ambil contoh terdapat gambar dengan ukuran 4x4 dan *kernel* berukuran 2x2 yang dinisiasi secara acak. Proses perhitungan operasi konvolusi dapat dilihat pada Gambar 3.



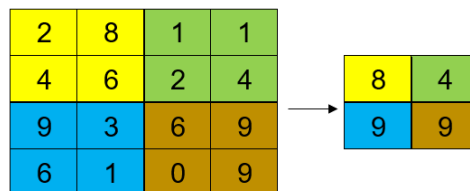
Gambar 3. Proses perhitungan operasi konvolusi

Pooling Layer. *Pooling* memiliki konsep untuk mengurangi ukuran dari gambar. Lapisan *pooling* akan mengurangi dimensi dan komputasi serta mengurangi kemungkinan terjadinya overfitting dikarenakan terdapat lebih sedikit parameter, model juga lebih toleran terhadap *variance* dan distorsi. *Average pooling* dan *max pooling* merupakan metode yang sering digunakan pada arsitektur CNN. *Average pooling* membagi input menjadi wilayah *pooling* berbentuk persegi dan menghitung nilai rata-rata dari setiap wilayah tersebut.



Gambar 4. *Average Pooling*

Sedangkan, *max pooling* akan mengambil nilai tertinggi dari wilayah yang dikonvolusi (Galety et al., 2021).

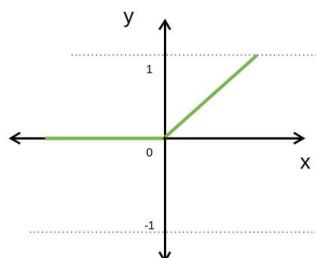


Gambar 5. *Max Pooling*

Rectifier Linear Unit (ReLU). Salah satu fungsi aktivasi yang sering digunakan di CNN adalah *Rectifier Linear Unit* (ReLU). Fungsi ini digunakan untuk mengonversi semua nilai masukan menjadi angka positif. ReLU memiliki keuntungan dalam komputasi karena memiliki beban yang sangat minimal dibandingkan dengan yang lain (Ghosh et al., 2019). Representasi matematika ReLU dapat dilihat pada persamaan (1.4) dan grafik fungsinya pada Gambar 6.

$$f(x)_{ReLU} = \max(0, x) \quad (1.4)$$

Dimana x merupakan nilai inputan ke *neuron*. Grafik fungsi ReLU dapat dilihat pada Gambar 6.



Gambar 6. Grafik Fungsi Aktivasi ReLU

Normalization Layer. Teknik normalisasi disebut sebagai komponen yang sangat efektif dalam *deep learning*. Metode normalisasi membantu pelatihan dari model *neural networks* dan meningkatkan performanya. Salah satu pendekatan normalisasi yang paling populer adalah *Batch Normalization* (BatchNorm atau BN) (Wang et al., 2020). Teknik *Batch Normalization* telah terbukti meningkatkan kecepatan pelatihan, memperbolehkan *learning rate* yang lebih besar, dan meningkatkan akurasi generalisasi model. *Batch Normalization* telah diterapkan sebagai unit dasar di arsitektur jaringan yang populer seperti ResNet dan *DenseNet*.

Fully Connected Layer. Dalam arsitektur klasifikasi CNN, *fully connected* (FC) *layer* biasanya terletak di bagian akhir dari arsitektur, dimana setiap *neuron* di dalam lapisan tersebut terhubung dengan setiap *neuron* dari lapisan sebelumnya. Lapisan FC mengambil input dari lapisan konvolusi atau pooling yang mengandung sekumpulan metrik (*feature maps*). Bagian terakhir dari *fully connected layer* digunakan sebagai *output layer* (*classifier*) (Ghosh et al., 2019).

1.6.4 Transfer Learning

Satu hal yang diinginkan dari mesin adalah mentransfer pengetahuan mereka dari suatu domain yang telah dilatih (*source domain*) ke domain lain (*target domain*) dimana data biasanya sedikit atau memerlukan pelatihan yang cepat (Neyshabur et al., 2020). *Transfer Learning* adalah metode yang memanfaatkan model yang telah dilatih sebelumnya (*pre-trained*) pada suatu dataset yang lain. Alur kerja yang paling umum dari penerapan transfer learning yaitu,

1. Mengambil lapisan-lapisan dari model yang telah dilatih sebelumnya
2. Membekukan lapisan-lapisan tersebut untuk mencegah hilangnya informasi selama pelatihan
3. Menambahkan lapisan baru di atas lapisan yang dibekukan dengan tujuan agar model dapat belajar mengubah fitur yang lama menjadi kemampuan prediksi pada dataset yang baru
4. Melatih lapisan baru tersebut menggunakan dataset yang baru.

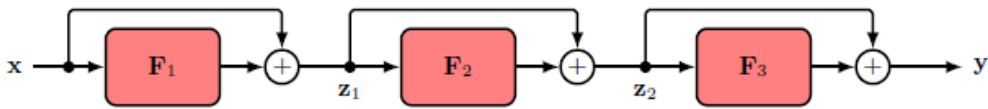
Transfer Learning memerlukan *fine-tuning* atau penyesuaian model *pre-trained* sesuai dengan permasalahan yang diinginkan (Wijaya et al., 2021).

1.6.5 Residual Network

Kedalaman jaringan dari *Neural Network* yang semakin bertambah memunculkan sebuah masalah yang tak terduga yaitu penurunan presisi pelatihan. Berbagai teknik telah dijelajahi untuk mengatasi masalah ini, seperti strategi inialisasi, teknik optimisasi, teknik normalisasi, dan *residual blocks*. Berdasarkan upaya yang telah dilakukan ditemukan bahwa *residual blocks* menunjukkan kinerja yang baik bahkan lebih baik dibandingkan teknik-teknik lainnya (Alaeddine dan Jihene, 2021).

Dalam *Residual block* terdapat komponen penting dalam mengatasi masalah degradasi yaitu *skip connection*. *Skip connection* merupakan komponen yang

memperbolehkan jaringan untuk melewati lapisan-lapisan tertentu dan menggunakan kembali *feature* dari lapisan sebelumnya (Alharthi dan Alzahrani, 2023).



Gambar 7. Jaringan *residual* (Bishop dan Bishop, 2024).

Untuk sebuah *neural network* yang terdiri dari tiga lapisan dengan bentuk sebagai berikut:

$$z_1 = F_1(x) \quad (1.5)$$

$$z_2 = F_2(z_1) \quad (1.6)$$

$$y = F_3(z_2) \quad (1.7)$$

Fungsi F_1 di sini dapat berupa transformasi linier diikuti oleh fungsi aktivasi ReLU atau bisa juga lebih kompleks dengan beberapa lapisan linier, fungsi aktivasi, dan lapisan normalisasi. Secara sederhana, koneksi *residual* menambahkan inputan untuk setiap fungsi ke output sehingga menghasilkan

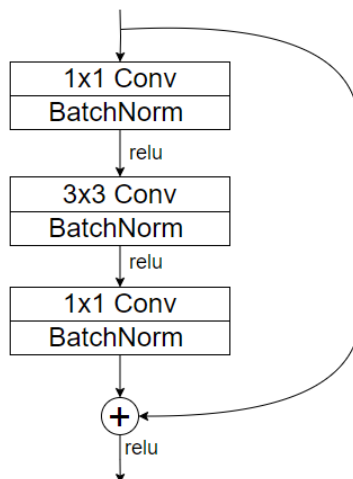
$$z_1 = F_1(x) + x \quad (1.8)$$

$$z_2 = F_2(z_1) + z_1 \quad (1.9)$$

$$y = F_3(z_2) + z_2 \quad (1.10)$$

Setiap kombinasi dari fungsi dan koneksi *residual*, seperti $F_1(x) + x$ disebut sebagai *residual block*. Jaringan yang tersusun atas beberapa *residual block* disebut sebagai *Residual Network* (ResNet) (Bishop dan Bishop, 2024).

Dalam ResNet-50, *residual block* yang digunakan disebut sebagai *bottleneck residual* yang dapat meningkatkan efisiensi komputasi secara besar-besaran dan secara signifikan meningkatkan kedalaman dari *residual block* (Sun et al., 2021). *Bottleneck residual* tersusun atas tiga lapisan konvolusi berukuran 1x1, 3x3, dan 1x1. Selain itu, *batch normalization* diterapkan setiap sesudah konvolusi dan sebelum aktivasi (He et al., 2016).



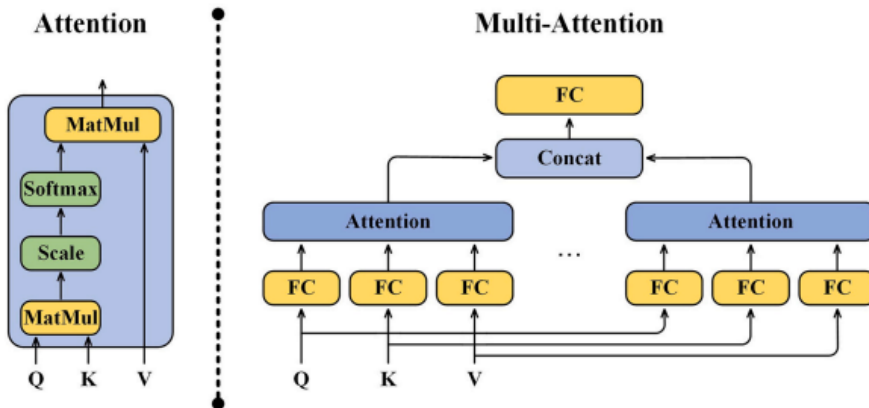
Gambar 8. *Bottleneck residual*

1.6.6 Vision Transformer

Pada mulanya, *Transformer* dikenalkan pada bidang *Natural Language Processing* (NLP) dengan didasarkan pada mekanisme *attention*. Baru-baru ini, arsitektur dari *transformer* mulai diterapkan pada tugas *computer vision* dan mendapatkan hasil yang tinggi. *Vision Transformer* (ViT) memberikan hasil yang menjanjikan serta dapat mengungguli performa dari CNN (Islam, 2022).

Vision Transformer bekerja dengan cara memecah gambar masukan menjadi beberapa bagian yang lebih kecil dimana setiap bagian akan direpresentasikan dalam bentuk vektor. Setiap bagian vektor akan diratakan menjadi rangkaian satu dimensi dan berperan sebagai masukan dari model *Transformer*. Selanjutnya, rangkaian ini akan diproses oleh *Transformer encoder* yang terdiri atas lapisan *self-attention* dan *Multi-Layer Perceptron* (MLP). Pengaturan ini dapat memperbolehkan model untuk menemukan perbedaan hubungan yang rumit antara bagian-bagian gambar yang berbeda (Rachman et al., 2024). Berikut ini penjelasan lebih lanjut mengenai *self-attention*.

Self-attention. Komponen kunci dari arsitektur *Transformer* adalah *self-attention* yang mengakibatkan model untuk fokus dan mengevaluasi berbagai hubungan yang signifikan antara berbagai elemen dan rangkaian data, seperti huruf-huruf atau bagian-bagian dari sebuah gambar (Rachman et al., 2024). Modul *attention* diekspresikan dengan bobot (*weight*) *attention* yang didapatkan dari menjumlahkan semua nilai bobot dari urutan (*sequence*). Mekanisme *attention* dari model *transformer* diimplementasikan *multi-head scaled dot-product* terhadap *query*, *key*, dan *value*.



Gambar 9. Alur diagram modul *attention* dan *multi-attention* (Peng & Wang, 2022).

Modul *attention* memiliki tiga input yaitu *queries* (Q), *keys* (K) dan *values* (V). Operasi *attention* dapat didefinisikan dalam operasi matematika:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1.11)$$

Input dari Q dan K memiliki dimensi d_k dan V memiliki dimensi d_v . Selanjutnya menentukan pentingnya setiap bagian dengan menghitung *dot product* dari Q dengan K, membagi hasilnya dengan $\sqrt{d_k}$ untuk menyesuaikan skala hasil dari *dot product*, kemudian menggunakan fungsi *softmax* untuk mendapatkan bobot pada V.

Agar model dapat menangkap berbagai jenis pengetahuan dari berbagai jenis perlakuan pada mekanisme *attention* yang sama dilakukan konversi Q, K, dan V menggunakan beberapa proyeksi linear yang dipelajari secara independen. Secara umum, proyeksinya dilakukan dengan menerapkan lapisan *fully-connected* (FC). Selanjutnya, operasi *attention* dilakukan secara paralel berdasarkan dari proyeksi Q, K, dan V sebelum akhirnya semua hasil keluaran *attention pooling* dihubungkan (*Concatenation*) dan ditransformasikan dengan proyeksi linear lain yang dipelajari (W^o) untuk menghasilkan keluaran akhir. Operasi ini disebut sebagai *multi-head attention*, dimana setiap kepala (*head*) akan mewakili satu *attention pooling*.

$$\text{MultiHeadAttn}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_i)W^o \quad (1.12)$$

dimana head_i :

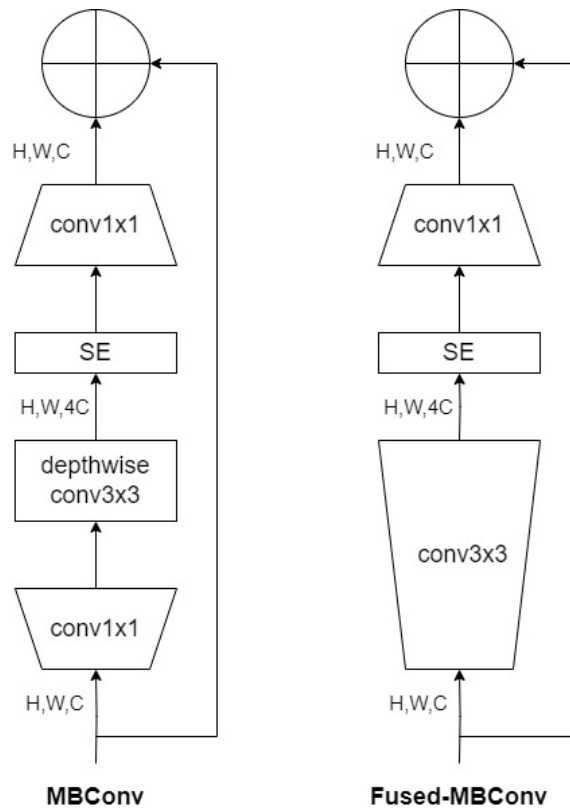
$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (1.13)$$

dimana W^o, W_i^Q, W_i^K, W_i^V merupakan matriks proyeksi yang dipelajari (Peng & Wang, 2022).

1.6.7 Efficient Network

EfficientNet merupakan model *Convolutional Neural Network* yang diusulkan untuk meningkatkan kinerja CNN dengan cara menskalakan tiga dimensi yaitu lebar (width), kedalaman (depth), dan resolusi (resolution) gambar input (Duong et al., 2020). Penskalaan ini dilakukan menggunakan metode *compound scaling* yang mengatur skala jaringan dengan serangkaian koefisien skala yang tetap. Metode *compound scaling* ini akan membantu untuk mengkoordinasikan dan menyeimbangkan penskalaan dimensi. *EfficientNet* berhasil mendapatkan kinerja dan komputasi yang lebih efisien ditandai dengan jumlah parameter dan FLOPs (Floating Point Operations per Second) yang lebih sedikit dibandingkan dengan model-model lainnya (Tan & Le, 2019).

Terlepas dari kesuksesan dari *EfficientNet*, terdapat beberapa masalah yaitu kecepatan *training* menjadi sangat lambat dan membutuhkan memori yang sangat besar ketika ukuran gambar menjadi lebih besar (Huang & Liao, 2023). Pada tahun 2021, Tan dan Le mengembangkan model jaringan *EfficientNetV2* yang berfokus untuk meningkatkan kecepatan *training* dan efisiensi parameter. Untuk mencapai hal tersebut, *EfficientNetV2* menggunakan kombinasi dari *training-aware neural architecture search* (NAS) dan *scaling*. NAS berfungsi untuk mencari kombinasi dari tipe operasi konvolusi (MBConv, Fused-MBConv), jumlah lapisan, ukuran kernel (3x3, 5x5) dan rasio ekspansi dengan tujuan untuk mendapatkan arsitektur yang optimal dalam klasifikasi gambar (Tan dan Le, 2021b). Struktur MBConv dan Fused-MBConv dapat dilihat pada Gambar 10.



Gambar 10. Struktur MBConv dan Fused-MBConv

1.6.8 Confusion Matrix

Confusion Matrix merupakan metode yang berperan dalam mengevaluasi performa model klasifikasi. Dalam menggunakan metode ini diperlukan untuk menguraikan jumlah nilai yang diprediksi terhadap jumlah nilai pada kelas yang benar dan kelas yang salah. Pada Tabel 1, T_i menunjukkan jumlah data prediksi yang sama dengan data sebenarnya dan $F_{(i,j)}$ menunjukkan jumlah data yang diprediksi sebagai kelas x tetapi data sebenarnya berada pada kelas y .

Tabel 1. Confusion Matrix

Actual Class	Predicted Class								
	Classes	Class 1	Class 2	Class 3	Class 4	...	Class 24	Class 25	Class 26
	Class 1	T_1	$F_{(1,2)}$	$F_{(1,3)}$	$F_{(1,4)}$	...	$F_{(1,24)}$	$F_{(1,25)}$	$F_{(1,26)}$
	Class 2	$F_{(2,1)}$	T_2	$F_{(2,3)}$	$F_{(2,4)}$	...	$F_{(2,24)}$	$F_{(2,25)}$	$F_{(2,26)}$
	Class 3	$F_{(3,1)}$	$F_{(3,2)}$	T_C	$F_{(3,4)}$	...	$F_{(3,24)}$	$F_{(3,25)}$	$F_{(3,26)}$
	Class 4	$F_{(4,1)}$	$F_{(4,2)}$	$F_{(4,3)}$	T_D	...	$F_{(4,24)}$	$F_{(4,25)}$	$F_{(4,26)}$
	:	:	:	:	:	...	:	:	:
	Class 24	$F_{(24,1)}$	$F_{(24,2)}$	$F_{(24,3)}$	$F_{(24,4)}$	...	T_{24}	$F_{(24,25)}$	$F_{(24,26)}$
	Class 25	$F_{(25,1)}$	$F_{(25,2)}$	$F_{(25,3)}$	$F_{(25,4)}$	...	$F_{(25,24)}$	T_{25}	$F_{(25,26)}$
	Class 26	$F_{(26,1)}$	$F_{(26,2)}$	$F_{(26,3)}$	$F_{(26,4)}$	...	$F_{(26,24)}$	$F_{(26,25)}$	T_{26}

Dengan informasi yang didapatkan dari Tabel 1, parameter pengujian lainnya seperti *Accuracy*, *Precision*, *Recall*, dan *F1-Score* dapat dihitung. Berikut penjelasan lebih lanjut mengenai parameter tersebut:

- a) *Accuracy* merupakan rasio prediksi seluruh klasifikasi yang benar terhadap keseluruhan data. Rumus *accuracy* dapat dituliskan:

$$Accuracy = \frac{\sum_{i=1}^n T_i}{N} \quad (1.14)$$

- b) *Precision* merupakan rasio prediksi klasifikasi yang benar positif terhadap keseluruhan data yang diprediksi positif. Rumus *precision* dapat dituliskan:

$$Precision_i = \frac{T_i}{T_i + \sum_{i=1}^n F_{ij}}, i \neq j \quad (1.15)$$

- c) *Recall* merupakan rasio prediksi klasifikasi yang benar positif terhadap keseluruhan data yang kelasnya positif. Rumus *recall* dapat dituliskan:

$$Recall_i = \frac{T_i}{T_i + \sum_{j=1}^n F_{ij}}, i \neq j \quad (1.16)$$

- d) *F1-Score* merupakan rata-rata harmonis antara *precision* dan *recall*. Rumus *F1-Score* dapat dituliskan:

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad (1.17)$$

Keterangan:

i = baris pada masing-masing *class*

j = kolom pada masing-masing *class*

n = jumlah *class*

N = jumlah keseluruhan data

BAB II

METODE PENELITIAN

2.1 Waktu dan Tempat Penelitian

2.1.1 Waktu Penelitian

Penelitian ini dilaksanakan dari bulan Juli 2024 sampai dengan bulan Oktober 2024. Adapun detail waktu penelitian dapat dilihat pada Tabel 2.

Tabel 2. Waktu penelitian

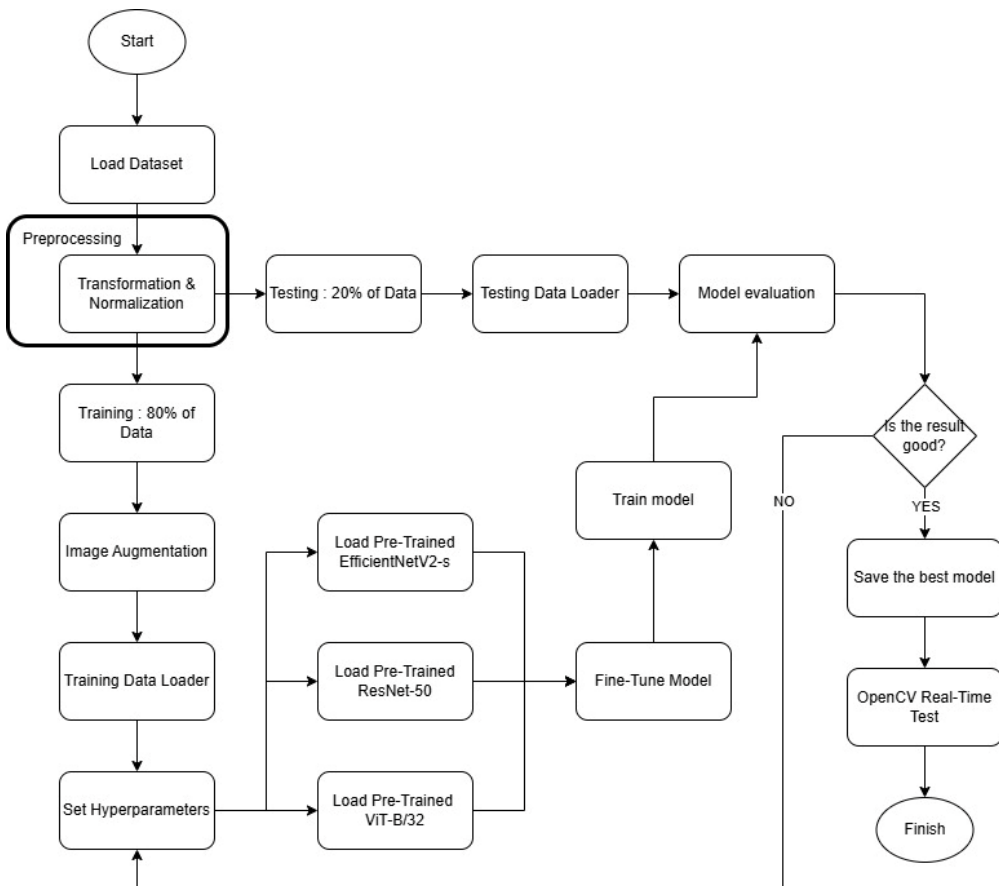
Kegiatan	Juli				Agustus					September				Oktober			
	1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	4
Studi Literatur																	
Data Collecting																	
Data Preprocessing																	
Build and Train Model																	
Model Evaluation																	
Real-Time Testing																	

2.1.2 Tempat Penelitian

Penelitian ini dilakukan di Laboratorium Rekayasa Perangkat Lunak, Program Studi Sistem Informasi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin.

2.2 Tahap Penelitian

Tahap pada penelitian ini merupakan langkah-langkah yang dilakukan untuk menyelesaikan penelitian. Tahapan yang dilalui untuk menyelesaikan penelitian ini meliputi: pengumpulan dataset, *preprocessing* data, augmentasi citra, mengatur *hyperparameter*, implementasi *transfer learning*, melatih model dan evaluasi kinerja model. Tahapan pada penelitian ini dapat dilihat pada Gambar 11.



Gambar 11. Alur Diagram Penelitian

2.2.1 Dataset

Penelitian ini menggunakan data primer yang dibuat oleh peneliti sendiri dan data sekunder yang diperoleh dari halaman *kaggle* dan *roboflow*. Data primer diambil menggunakan kamera *smartphone* dengan *aspect ratio* 1:1 di dalam ruangan di bawah lampu pijar. Nama dataset dan sumber pengambilannya dapat dilihat pada Tabel 3.

Tabel 3. Sumber Data

Nama dataset	Jumlah Kelas	Sumber data
Data Primer BISINDO	26 Kelas (1248 Citra)	Dibuat oleh peneliti
BISINDO revisi	26 Kelas (104 Citra)	https://universe.roboflow.com/ikado/bisindo-revisi
Bahasa Isyarat Indonesia (BISINDO) <i>Alphabets</i>	26 Kelas (312 Citra)	https://www.kaggle.com/datasets/achmadnoer/alfabet-bisindo
BISINDO <i>Letter</i> Dataset	26 Kelas (936 Citra)	https://www.kaggle.com/datasets/alfredolorentiars/bisindo-letter-dataset

Empat dataset tersebut kemudian akan digabungkan dan disebut *Merged Dataset*. Format citra yang terdapat pada *Merged Dataset* adalah .jpg dan .png dengan total 2600 citra.

2.2.2 Preprocessing

Preprocessing data yang diterapkan pada data citra huruf alfabet meliputi transformasi dan normalisasi. Ukuran citra pada *datatrain* dan *datatest* akan diubah dengan ukuran 3x224x224. Setelah itu data citra akan dikonversi ke format *tensor* dan nilai piksel citra akan diskalakan dari rentang 0-255 menjadi rentang 0-1. Terakhir, data *tensor* citra dinormalisasikan dengan rata-rata dan standar deviasi.

2.2.3 Split Data

Proses selanjutnya adalah membagi data menjadi data *training* dan data *testing*. Sebanyak 80% dari keseluruhan data akan digunakan sebagai data *training* dan 20% lainnya akan digunakan sebagai data *testing*.

2.2.4 Augmentasi Citra

Augmentasi citra dilakukan untuk mengurangi terjadinya *overfitting* dengan menambah variasi data. Metode augmentasi yang digunakan pada penelitian ini berupa pembalikan horizontal, mengubah kecerahan gambar, dan memberikan efek blur. Proses augmentasi citra hanya dilakukan pada data *training*.

2.2.5 Data Loader dan Hyperparameter

Data loader bertujuan mempercepat proses pelatihan model dan mengurangi terjadinya *overfitting* pada model dengan menerapkan *batch size* dan *reshuffle* data. Nilai *batch size* yang digunakan adalah 32 dan *reshuffle* data hanya dilakukan pada data *training*. Selain itu, *epoch* yang digunakan sebanyak 25, *learning rate* sebesar 0,001 dengan *weight decay* sebesar 0,0001.

2.2.6 Transfer Learning dan Fine-Tune Model

Transfer learning yang dilakukan pada penelitian melalui tahap sebagai berikut;

- a) *Load pretrained* model dan atur bobot menjadi "DEFAULT" untuk mendapatkan bobot yang digunakan pada pelatihan dataset *ImageNet-1K*.
- b) Bekukan (*freeze*) seluruh lapisan dari *pretrained network* kecuali lapisan terakhir untuk menjaga *value* dari *pretrained* tetap terjaga.
- c) Ganti lapisan terakhir dengan *sequence* yang memiliki lapisan *dropout* untuk mengurangi terjadinya *overfitting* dan lapisan *linear* yang memiliki jumlah kelas yang sesuai dari jumlah kelas huruf alfabet bahasa isyarat Indonesia (BISINDO).
- d) Definisikan fungsi *loss* dan optimisasi yang digunakan.
- e) Selanjutnya latih model dan lakukan evaluasi untuk mengetahui kinerja model.

2.2.7 Evaluasi

Evaluasi dilakukan melalui dua cara yaitu pengujian menggunakan data *testing* dan pengujian menggunakan input video secara *real-time*. Pengujian menggunakan data *testing* bertujuan untuk mengetahui kinerja setiap model dengan melihat *confusion matrix*, akurasi, *loss*, *presisi*, *recall*, dan *f1-score*. Sedangkan, pengujian menggunakan input video secara *real-time* bertujuan untuk mengetahui kinerja model dalam menghadapi *noise* ketika diterapkan pada situasi dunia nyata.

2.3 Instrumen Penelitian

Perangkat keras (*Hardware*) yang digunakan pada penelitian ini yaitu laptop dengan *processor Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz*, RAM 12 GB, kartu grafis NVIDIA GeForce MX330 2GB, dan sistem operasi Windows 11. *Library* yang digunakan untuk penelitian ini adalah *torch*, *sklearn*, *matplotlib*, dan *OpenCV* dengan bahasa pemrograman *python*. Untuk menjalankan kode, digunakan *Kaggle Notebook* dengan akselerator GPU T4 x2. Sedangkan, pembuatan *environment* untuk *real-time testing* menggunakan *Jupyter Notebook*.