

## Penerapan Convolutional Neural Network Pada Aplikasi Terjemahan Bahasa Isyarat Indonesia Berbasis Android

Wulan Zahra Putri<sup>1</sup>, Irma Salamah<sup>1\*</sup>, Emilia Hesti<sup>1</sup>

<sup>1</sup>Program Studi Teknik Telekomunikasi, Politeknik Negeri Sriwijaya, Indonesia.

---

### Artikel Info

#### Kata Kunci:

Android;  
Bahasa Isyarat Indonesia;  
Convolutional Neural Network;  
MobileNetV2.

#### Keywords:

Android;  
Indonesian Sign Language;  
Convolutional Neural Network;  
MobileNetV2.

#### Riwayat Artikel:

Submitted: 03 Juli 2025  
Accepted: 31 Juli 2025  
Published: 31 Juli 2025

**Abstrak:** Pentingnya bahasa isyarat terletak pada perannya sebagai sistem komunikasi visual-manual, di mana isyarat tangan, mimik wajah, dan gerakan badan digunakan untuk menyampaikan informasi. Sistem ini terutama digunakan oleh individu tunarungu sebagai metode komunikasi utama mereka. Walaupun krusial, akses yang terbatas terhadap sumber belajar sering kali menjadi kendala signifikan dalam upaya mempelajari Bahasa Isyarat Indonesia (BISINDO) secara efektif. Menanggapi tantangan ini, penelitian ini dilakukan dengan tujuan mengembangkan aplikasi terjemahan bahasa isyarat melalui integrasi Convolutional Neural Network (CNN), diterapkan secara spesifik melalui arsitektur MobileNetV2. Dalam proses pengembangannya, aplikasi ini menggunakan dataset yang terdiri dari 2000 video, yang selanjutnya diubah menjadi 48.000 bingkai gambar untuk pelatihan model. Pengujian yang dilakukan menunjukkan bahwa akurasi pengenalan gerakan tangan bahasa isyarat mencapai 76%, dengan total loss tercatat sebesar 0.81. Data ini mengindikasikan bahwa model menunjukkan performa yang memadai dalam mengklasifikasikan mayoritas data pelatihan. Selain fitur deteksi gerakan tangan untuk bahasa isyarat, aplikasi ini juga memiliki fitur terjemahan teks ke isyarat, yang dapat mengonversi kata menjadi gambar gerakan bahasa isyarat. Selanjutnya, terdapat fitur kamus yang menyajikan daftar kata-kata, masing-masing dilengkapi dengan video demonstrasi gerakan bahasa isyarat yang relevan. Fitur-fitur ini bertujuan untuk memfasilitasi dan mempercepat proses pembelajaran BISINDO, menjadikannya mudah diakses bagi individu yang ingin menguasainya.

**Abstract:** The importance of sign language lies in its role as a visual-manual communication system, where hand gestures, facial expressions, and body movements are used to convey information. This system is used by deaf individuals as their primary method of communication. Although crucial, limited access to learning resources is often an obstacle in learning Indonesian Sign Language (BISINDO). In response to this challenge, this study was conducted with the aim of developing a sign language translation application through the integration of Convolutional Neural Network (CNN), implemented through the MobileNetV2 architecture. In the development process, this application used a dataset consisting of 2000 videos, which were then converted into 48,000 image frames for model training. Testing showed that the accuracy of sign language hand gesture recognition reached 76%, with a total loss recorded at 0.81. This data indicates that the model shows adequate performance in classifying the majority of the training data. In addition to the hand gesture detection feature for sign language, this application also has a text-to-sign translation feature, which can convert words into sign language gesture images. Next, there is a dictionary feature that presents a list of words, each accompanied by a video demonstration of the relevant sign language movements. These

---

*features aim to facilitate the learning of BISINDO, making it easily accessible for individuals who want to master it.*

---

**Corresponding Author:**

Irma Salamah

Program Studi Teknik Telekomunikasi, Politeknik Negeri Sriwijaya.

Alamat: Jl. Srijaya Negara, Bukit Lama, Kec. Ilir Bar. I, Kota Palembang, Sumatera Selatan, Indonesia

Email: [irma\\_salamah@polsri.ac.id](mailto:irma_salamah@polsri.ac.id)

---

## PENDAHULUAN

Komunikasi adalah proses transmisi pesan, gagasan, atau ide dari satu individu ke individu lainnya, yang disampaikan secara lisan ataupun verbal sehingga bisa dipahami berbagai individu (Dewi et al., 2021). Dalam komunikasi, elemen bahasa sangat krusial, termasuk bagi individu tunarungu. Dalam proses komunikasi, individu tunarungu membutuhkan bahasa sehari-hari yang mudah dipahami dan digunakan agar bisa berkomunikasi secara efektif, baik untuk menyampaikan maupun memahami pesan (Nugraheni et al., 2023). Komunikasi ini umumnya dikenal sebagai bahasa isyarat.

Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO) merupakan dua bentuk utama bahasa isyarat di Indonesia. SIBI mencakup 26 gerakan jari yang masing-masing mewakili 26 huruf alfabet, dilakukan dengan satu tangan. Terdapat 24 gerakan statis dan 2 gerakan dinamis dalam sistem ini (Nurhayati et al., 2022), sedangkan Bahasa Isyarat Indonesia (BISINDO) yang awalnya diinisiasi oleh Gerakan untuk Kesejahteraan Tuna Rungu Indonesia (Gerkatina) dan terus berkembang melalui interaksi di antara anggota komunitas tunarungu (Borman et al., 2018).

Berdasarkan survei dan wawancara yang dilakukan, mayoritas individu tunarungu, tepatnya 91% dari 100 responden di berbagai daerah telah menggunakan BISINDO untuk berkomunikasi. Para penyandang tunarungu juga mendukung penggunaan BISINDO sebagai alat komunikasi mereka (Nasha Hikmatia A.E. & Zul, 2021), dikarenakan BISINDO menggunakan kedua tangan sehingga penyampaian menjadi lebih mudah dipahami. Maka dari itu, pada pengembangan aplikasi ini akan menggunakan BISINDO.

Seiring dengan kemajuan teknologi yang pesat, khususnya dalam bidang *Machine Learning*, terdapat sejumlah penelitian yang telah mengeksplorasi topik serupa. Berdasarkan penelitian yang dilaksanakan oleh Reza Haris Alfikri, Mardi Siswo Utomo, Herny Februariyanti, dan Eko Nurwahyudi dengan judul "PEMBANGUNAN APLIKASI PENERJEMAH BAHASA ISYARAT DENGAN METODE CNN BERBASIS ANDROID", mengembangkan sebuah aplikasi Android yang memiliki kemampuan menerjemahkan bahasa isyarat, didukung oleh teknologi *Convolutional Neural Network* (CNN). Penelitian ini mengungkapkan bahwa aplikasi tersebut mampu menerjemahkan bahasa isyarat dengan tingkat akurasi 73% setelah proses *deployment*, meskipun model awalnya menunjukkan akurasi hingga 95% selama fase pelatihan. Dataset yang digunakan mencakup 1820 gambar bahasa isyarat *American Sign Language* (ASL). Model dilatih selama 3 total epochs, menghasilkan akurasi 95% untuk 100 epoch, 93% untuk 150 epoch, dan 97% untuk 200 epoch. Aplikasi ini memiliki kelemahan, yaitu penurunan akurasi setelah *deployment* dan kesalahan prediksi pada beberapa isyarat, yang menyoroti pentingnya pemahaman pengguna terhadap bahasa isyarat (Alfikri et al., 2022).

Kemudian dilakukan juga penelitian oleh Lukman Arisandi dan Barka Satya dengan judul "Sistem Klarifikasi Bahasa Isyarat Indonesia (BISINDO) Dengan Menggunakan Algoritma *Convolutional Neural Network*", melakukan perancangan sistem klasifikasi yang memanfaatkan arsitektur CNN untuk pengenalan alfabet BISINDO berbasis MobileNetV2. Model yang dihasilkan menunjukkan akurasi 99.82% selama pengujian, yang mencerminkan kemampuan yang signifikan dalam mengenali 26 huruf BISINDO. Menggunakan dataset yang terdiri dari 2163 gambar yang terbagi ke dalam 9 kelas, dan model dilatih selama 1734 epoch dengan penerapan *callback* untuk menghindari *overfitting*.

Walaupun hasil yang di-peroleh menunjukkan potensi yang baik, terdapat beberapa kelemahan dalam penelitian ini. Salah satu poin yang butuh pendalaman adalah keterbatasan jumlah kelas yang hanya berjumlah 9. Di samping itu, kurangnya pengujian dalam kondisi real-time juga menjadi aspek penting yang diperlukan untuk penerapan praktis dalam kehidupan sehari-hari (Arisandi & Satya, 2022).

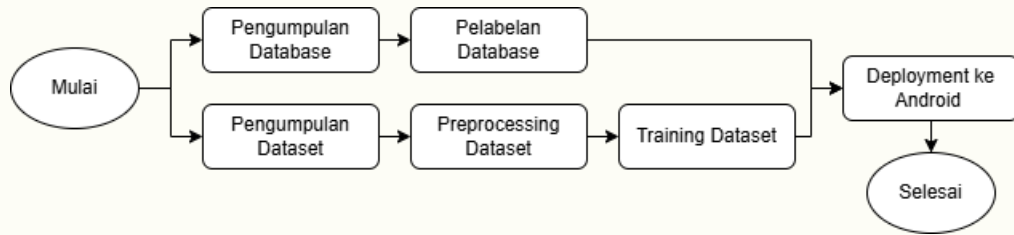
Terdapat penelitian juga yang dilakukan oleh Mughaffir Yunus dan Yusril Anwar dengan judul “APLIKASI PENERJEMAH BAHASA ISYARAT INDONESIA KE DALAM HURUF ABJAD”, mengembangkan sebuah aplikasi yang berfungsi untuk menerjemahkan Bahasa Isyarat Indonesia ke dalam huruf abjad, dengan memanfaatkan metode *Convolutional Neural Network* (CNN) yang diimplementasikan melalui model MobileNetV2. Dataset yang dianalisis berjumlah 11.144 gambar yang terklasifikasi ke dalam 27 kategori, yaitu huruf abjad. Proses pelatihan berlangsung selama 60 epoch dengan laju pembelajaran 0.001 dan ukuran batch 16. Pengujian menghasilkan akurasi sebesar 0.9995 dan nilai loss tercatat pada 0.6982. Meskipun memiliki tingkat akurasi yang tinggi, aplikasi ini menunjukkan kelemahan dalam performa yang mungkin tidak optimal ketika diterapkan dalam kondisi dunia nyata, khususnya terkait dengan variasi latar belakang dan jarak dari kamera (Yunus & Anwar, 2022).

Selanjutnya, penelitian yang dilakukan oleh Reihan Saputra, Gentur Wahyu Nyipto Wibowo, dan Akh-mad Khanif Zyen dengan judul “Sistem Klasifikasi Alfabet Bahasa Isyarat Indonesia Menggunakan CNN dengan MobileNetV2 berbasis Android”, yang mengembangkan aplikasi untuk mengklasifikasikan alfabet Bahasa Isyarat Indonesia (BISINDO) menggunakan kamera *smartphone* secara *real-time*. Memanfaatkan dataset yang terdiri dari 5200 gambar alfabet BISINDO, dengan 200 gambar per kelas (26 kelas). Model dibangun menggunakan MobileNetV2 dengan transfer learning dari ImageNet, dioptimalkan menggunakan Adam dengan learning rate  $1e-4$ , dan diterapkan teknik augmentasi data. Hasil evaluasi menunjukkan akurasi validasi sebesar 95% dan akurasi pengujian 93,85%. Kelemahan dari penelitian ini adalah kemungkinan kesalahan klasifikasi untuk huruf yang mirip secara visual dan terbatasnya kemampuan pengenalan gambar statis dalam situasi dunia nyata (Saputra et al., n.d.).

Penelitian ini dilakukan dengan tujuan mengembangkan aplikasi berbasis android yang dapat menerjemahkan bahasa isyarat memanfaatkan *Convolutional Neural Network* (CNN), terkhususnya arsitektur MobileNetV2. *Convolutional Neural Network* (CNN) adalah proses transformasi data input melalui konvolusi yang dilakukan secara berulang dan berurutan pada berbagai skala spasial, contohnya melalui operasi penggabungan (Kattenborn et al., 2021). *Convolutional Neural Network* (CNN) berfungsi sebagai alat klasifikasi untuk data yang telah memiliki label, memanfaatkan pendekatan *supervised learning* (Prasiwinigrum & Lubis, 2024). Aplikasi ini dikembangkan sebagai sarana edukasi yang dapat mendukung individu tunarungu serta masyarakat umum yang ingin memahami dan mempelajari Bahasa Isyarat Indonesia (BISINDO). Dengan memanfaatkan teknologi pengenalan pola, aplikasi ini diharapkan dapat mengenali gerakan tangan, sehingga memungkinkan pengguna untuk lebih mudah memahami dan menggunakan bahasa isyarat. Selain itu, terdapat fitur terjemahan yang memungkinkan pengguna untuk memasukkan teks yang akan menghasilkan gambar bahasa isyarat dan fitur kamus yang berisi kumpulan kata yang hasilnya adalah video gerakan bahasa isyarat.

## METODE PENELITIAN

Perancangan sistem adalah fase penting dalam penelitian ini untuk menciptakan aplikasi pembelajaran bahasa isyarat yang efektif dan tepat. Diagram alir perancangan sistem yang telah disusun menyajikan gambaran yang jelas dan terstruktur mengenai alur kerja yang dilakukan. Diagram yang disajikan di bawah ini merinci setiap tahapan dengan sistematis, dimulai dari proses pengumpulan data awal, pengolahan data, pelatihan model *machine learning*, hingga tahap akhir yaitu implementasi atau *deployment* model ke dalam aplikasi Android. Setiap langkah dalam alur perancangan ini akan dijelaskan secara rinci sebagai berikut.



Gambar 1. Alur Penelitian

Dengan memanfaatkan arsitektur MobileNetV2, penelitian ini mengimplementasikan *Convolutional Neural Network* untuk pengenalan citra gerakan tangan Bahasa Isyarat Indonesia (BISINDO). Aplikasi ini menyajikan tiga fitur, yaitu Terjemahan, Kamus, dan Latihan. Dalam fitur Terjemahan, pengguna dapat memasukkan kata yang akan menampilkan gambar gerakan bahasa isyarat. Selanjutnya, fitur Kamus berisi kumpulan kata-kata yang hasilnya berupa video gerakan bahasa isyarat. Fitur terakhir adalah Latihan, yang dapat mengonversi gerakan tangan menjadi kata-kata dalam bahasa isyarat secara real-time. Tabel 1 menyajikan 40 kata yang digunakan dalam pengembangan aplikasi ini.

Tabel 1. Kata Yang Digunakan

Kata Yang Digunakan			
Apa Kabar	Duduk	Marah	Sedih
Apa	Halo	Melihat	Selamat Malam
Bagaimana	Kalian	Membaca	Selamat Pagi
Baik	Kami	Menulis	Selamat Siang
Belajar	Kamu	Mereka	Selamat Sore
Berapa	Kapan	Minum	Senang
Berdiri	Kemana	Pendek	Siapa
Bingung	Kita	Ramah	Terima Kasih
Dia	Makan	Sabar	Tidur
Dimana	Mandi	Saya	Tinggi

## 1. Pengumpulan Dataset

Pada pengembangan aplikasi ini, langkah yang pertama dilakukan adalah pengumpulan datasets yang terdiri dari kumpulan video yang berisikan gerakan bahasa isyarat. Dataset yang telah dikumpulkan akan berfungsi sebagai *input* untuk model *machine learning* yang nantinya akan dilatih dan diproses oleh sistem. Dataset yang digunakan berasal dari situs online yaitu Kaggle pada URL <https://www.kaggle.com/datasets/anggiyohanespardede/bisindo-40-kata-mp4>. Kaggle adalah Sebuah platform komunitas untuk ilmuwan data, Kaggle menyediakan ribuan dataset terbuka dari berbagai disiplin, termasuk pengenalan gambar, prediksi data tabular, dan pemrosesan teks (Peran et al., 2024). Gambar 2 menampilkan beberapa sampel dataset yang akan digunakan.



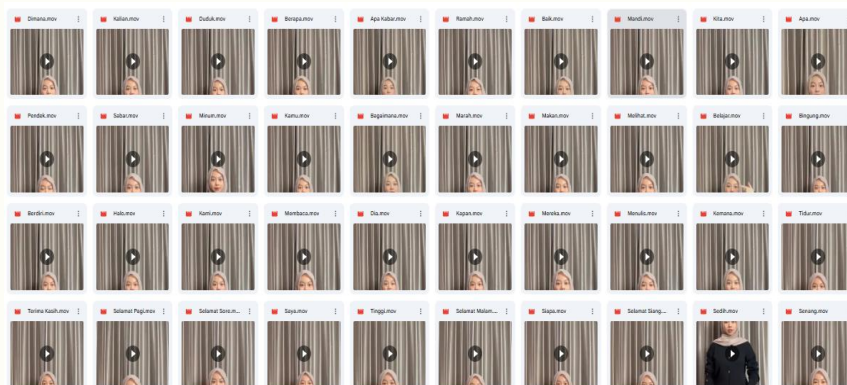
Gambar 2. Sampel Dataset



Dataset yang digunakan adalah Bahasa Isyarat Indonesia (BISINDO) yang diunggah oleh Anggi Yohanes Pardede pada tahun 2024. Dataset ini terdiri dari 40 kata, di mana setiap kata dilengkapi dengan 50 video, menghasilkan total 2000 video dan 48000 frame, dengan ukuran masing-masing 400 x 300.

## 2. Pengumpulan Database

Langkah kedua dalam pengembangan aplikasi ini adalah pengumpulan *database*. *Database* berfungsi sebagai repositori referensi yang digunakan saat aplikasi berjalan (*runtime*) untuk fitur Terjemahan dan Kamus. *Database* memberikan kemudahan signifikan dalam pengolahan data, khususnya dalam aspek pencarian dan pengambilan data. *Database* ini mencakup 40 gambar dan 40 video gerakan bahasa isyarat dengan ukuran 720 x 1280 yang akan digunakan sebagai *input* untuk fitur terjemahan yang menampilkan gambar gerakan bahasa isyarat dan fitur kamus yang akan menampilkan video gerakan bahasa isyarat. Gambar 3 menampilkan beberapa sampel *database* yang digunakan.



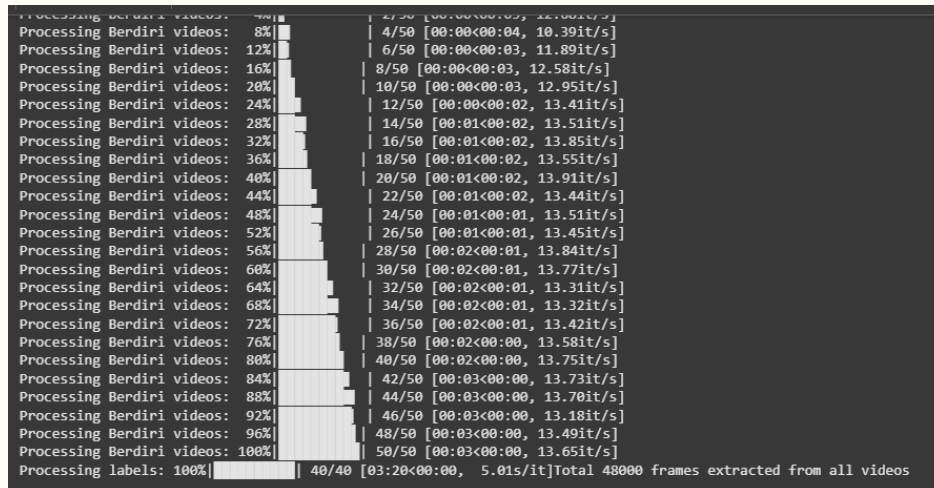
Gambar 3. Sampel Database

## 3. Pelabelan Database

Setelah database terkumpul, maka dilakukan pelabelan database yang akan digunakan sebagai referensi untuk menentukan kolom-kolom label dalam tabel database di MySQL. MySQL memiliki fungsi untuk mengelola dan menciptakan basis data di sisi server yang menyimpan berbagai informasi menggunakan SQL. MySQL adalah sistem yang bersifat *open source* dan mendukung integrasi dengan bahasa pemrograman lain. Sistem ini tidak memerlukan RAM yang besar dan juga mendukung penggunaan oleh banyak pengguna secara bersamaan (Bell, 2023). Dalam ilmu data, MySQL digunakan untuk menjalankan kueri pada dataset besar, sehingga mempermudah pengguna dalam mengakses data tersebut (Wahyudi et al., 2022).

## 4. Preprocessing Dataset

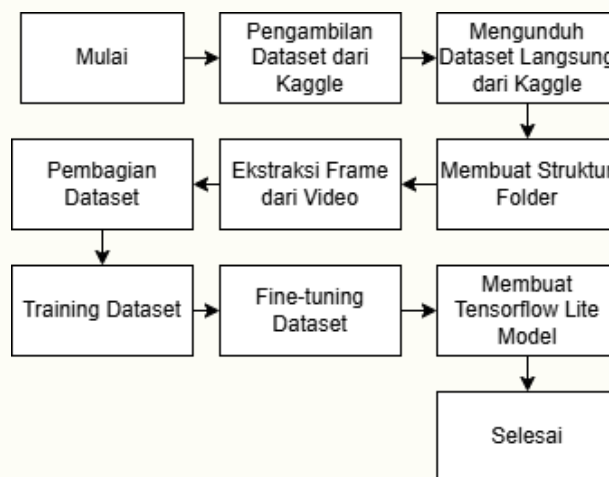
*Preprocessing* dataset merupakan tahap awal dalam pembelajaran mesin yang mengonversi dataset menjadi format yang dapat dipahami oleh model (Maharana et al., 2022). *Preprocessing* bertujuan untuk mengoptimalkan kualitas data masukan, sehingga meningkatkan efektivitas dan kinerja model dalam fase pelatihan (Sutarti & Syaqqalloh, 2025). Pada tahap ini, video dari Kaggle dipersiapkan untuk menjadi input dataset yang sesuai dengan metode CNN, guna menghasilkan model *machine learning* yang akurat. Proses ini mencakup ekstraksi 40 video dataset menjadi total 48.000 frame gambar. Parameter yang digunakan dalam *preprocessing* ini meliputi normalisasi ukuran gambar ke dimensi seragam seperti 224x224 piksel untuk memastikan konsistensi input bagi model CNN. Selain itu, konversi warna dilakukan dalam format RGB (Red, Green, Blue), mempertahankan informasi warna penuh yang penting untuk pengenalan gerakan isyarat. Gambar 4 akan menampilkan alur lengkap proses *preprocessing* dataset yang dijelaskan di atas.

Gambar 4. Proses *Preprocessing* Dataset

## 5. Training Dataset

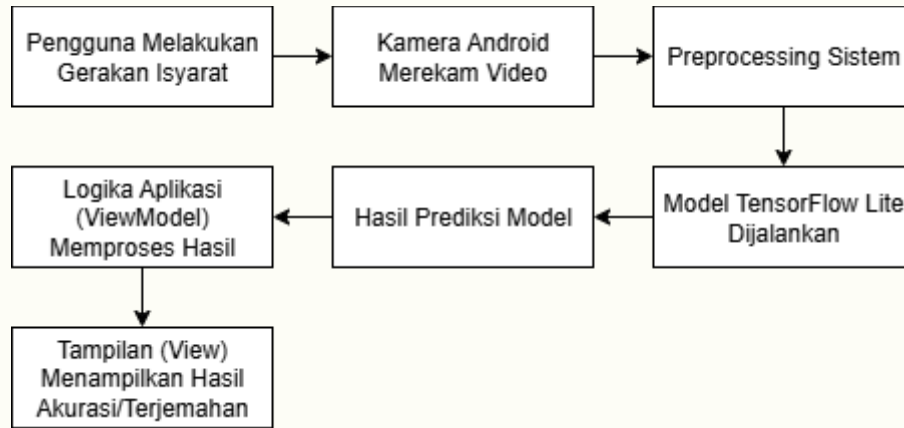
Tahap pelatihan (*training*) merupakan fase krusial dalam pengembangan model, di mana dataset diolah untuk mencapai akurasi tinggi dalam klasifikasi. Pada fase ini, dataset akan melalui proses pengujian menggunakan 40 epoch dengan metode *Convolutional Neural Network* (CNN). Arsitektur yang dipilih adalah MobileNetV2 yang merupakan turunan dari *convolutional neural network* yang didesain spesifik pada perangkat *mobile* agar dapat mengatasi keterbatasan sumber daya komputasi (Nasha Hikmatia A.E. & Zul, 2021). MobileNetV2 digunakan untuk melatih model agar mampu mengenali pola dari frame gambar yang diekstrak dari video hasil *preprocessing*.

Selama proses pelatihan, beberapa parameter kunci dikonfigurasi untuk mengoptimalkan kinerja model. `img_size` ditetapkan pada (224, 224), menentukan dimensi setiap gambar input yang diproses oleh model, sebuah ukuran standar untuk MobileNetV2 yang memastikan konsistensi data. `batch_size` diatur menjadi 32, mengelola jumlah sampel gambar yang diproses dalam satu iterasi sebelum bobot model diperbarui, menyeimbangkan kecepatan pelatihan dengan stabilitas gradien. Selain itu, `learning_rate` disetel ke  $1 \times 10^{-4}$ , sebuah tingkat pembelajaran yang relatif kecil untuk mencegah overfitting dan memungkinkan model menyempurnakan bobotnya secara bertahap. Sebagai pengoptimal, digunakan algoritma Adam (*Adaptive Moment Estimation*). Adam dipilih karena efisiensinya dalam secara adaptif menyesuaikan tingkat pembelajaran untuk setiap parameter model, yang berkontribusi pada konvergensi lebih cepat dan peningkatan kinerja model secara keseluruhan. Gambar 5 akan menampilkan alur lengkap proses pelatihan dataset hingga menghasilkan model TensorFlow Lite yang siap diimplementasikan pada aplikasi Android.

Gambar 5. Alur *Training* Dataset

## 6. Deployment ke Android

Saat training model selesai dilakukan, maka akan menghasilkan sebuah file model yang akan diintegrasikan ke aplikasi android pada Android Studio dengan menggunakan *library tensorflow* dan bahasa pemrograman kotlin. Kotlin merupakan bahasa pemrograman dengan tipe statis yang beroperasi pada *Java Virtual Machine* (JVM) dan sepenuhnya kompatibel dengan Java (Oliveira et al., 2020).

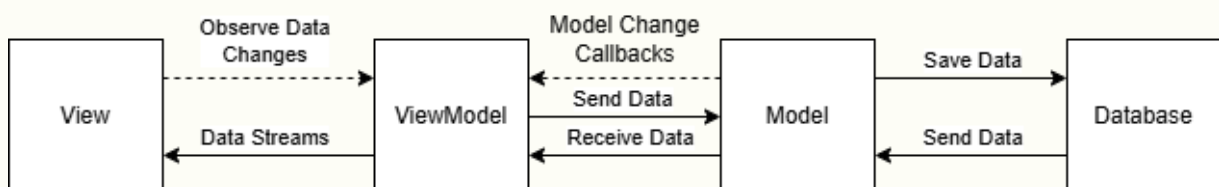


Gambar 6. Alur Integrasi Model Machine Learning

Gambar 6 menampilkan alur integrasi model ke fungsi Android yang dimulai ketika pengguna melakukan gerakan isyarat yang langsung direkam oleh kamera perangkat Android, menangkap input video atau frame gambar. *Frame* yang telah ditangkap kemudian melalui *preprocessing* seperti normalisasi ukuran dan konversi warna agar sesuai dengan input yang dibutuhkan model. Selanjutnya, *frame* yang sudah diproses ini dimasukkan ke dalam model *TensorFlow Lite* yang terintegrasi, yang kemudian melakukan inferensi untuk mengenali gerakan isyarat dan menghasilkan prediksi gerakan isyarat yang dikenali. Logika aplikasi (*ViewModel*) akan memproses hasil prediksi ini, baik untuk menghitung akurasi pada fitur Latihan dan mencocokkan kata yang sesuai. Terakhir, tampilan (*View*) akan memperbarui antarmuka pengguna untuk menyajikan hasil akhir kepada pengguna, seperti kata yang dimaksud dan akurasi pengenalan.

Aplikasi ini memiliki beberapa fitur, yaitu Terjemahan, Latihan, dan Kamus. Pada fitur Latihan akan menampilkan hasil akurasi pengenalan gerakan tangan bahasa isyarat dari model yang telah diintegrasikan ke dalam Android. Fitur Terjemahan akan mengambil hasil dari database yang telah disimpan di MySQL. Sedangkan pada fitur Kamus menyediakan daftar kata yang menampilkan video gerakan bahasa isyarat melalui integrasi dengan YouTube. Oleh karena itu, pengguna harus terhubung secara daring untuk mengakses aplikasi ini.

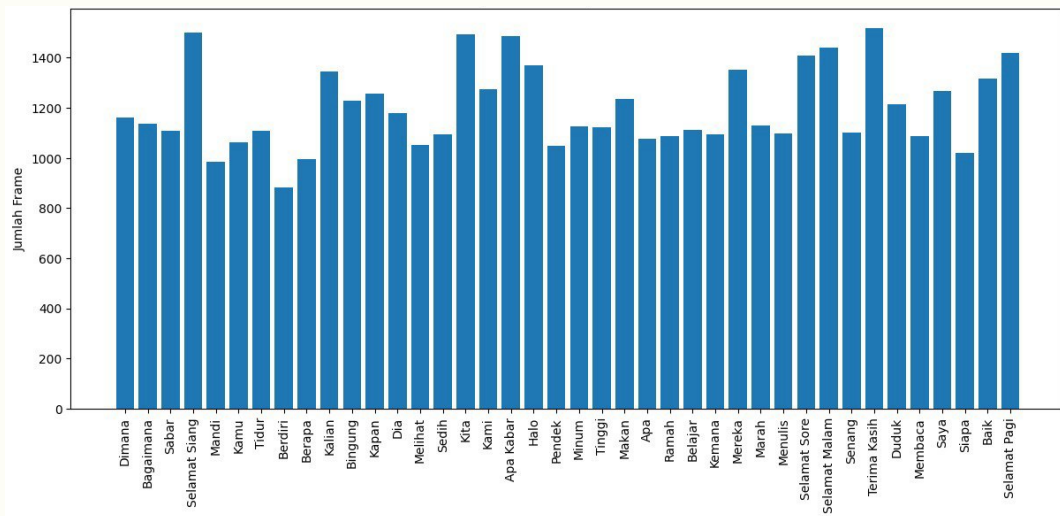
Pola Arsitektur MVVM (*Model-View-ViewModel*) yang akan digunakan dalam pengembangan aplikasi ini. Arsitektur MVVM memisahkan *View* sebagai tampilan antarmuka pengguna (UI), *ViewModel* sebagai penghubung antara *View* dan *Model* serta sebagai logika bisnis aplikasi, dan *Model* yang bertanggung jawab mengelola data aplikasi (Riyadhi et al., 2023). Gambar 7 menampilkan cara kerja dari arsitektur MVVM pada pengembangan aplikasi ini.



Gambar 7. Arsitektur MVVM

## HASIL DAN PEMBAHASAN

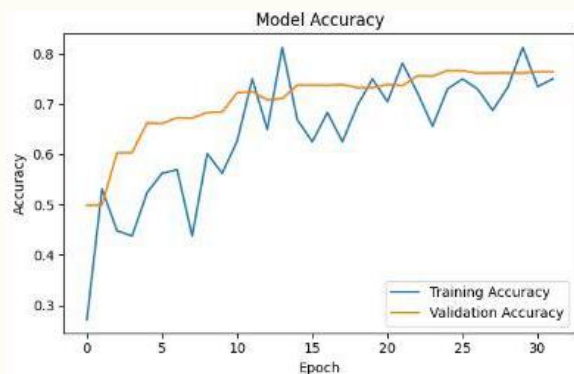
### 1. Hasil Preprocessing Dataset



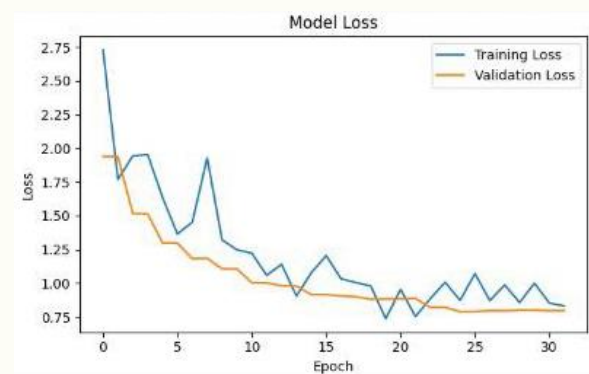
Gambar 8. Hasil Preprocessing Dataset

Pada Gambar 8 menyajikan hasil *preprocessing* dataset yang menunjukkan distribusi jumlah *frame* yang diekstraksi untuk setiap kata atau frasa dalam Bahasa Isyarat Indonesia (BISINDO). Observasi menunjukkan variasi dalam jumlah *frame* per kelas, dengan mayoritas kata/frasa memiliki lebih dari 1000 *frame*, yang mengindikasikan kecukupan volume data untuk pelatihan model *deep learning*, khususnya MobileNetV2.

### 2. Hasil Training Dataset



Gambar 9. Hasil Akurasi Model



Gambar 10. Hasil Total Loss

Berdasarkan pengujian yang dilakukan, penelitian ini berhasil mengimplementasikan arsitektur MobileNetV2 yang dilatih dengan dataset berjumlah 48.000 frame video. Hasil pelatihan menunjukkan akurasi sebesar 76%, yang mengindikasikan bahwa model dapat mengklasifikasikan sebagian besar data pelatihan dengan tepat, sedangkan total loss 0.81 mencerminkan tingkat kesalahan prediksi yang masih ada, menandakan potensi peningkatan kinerja.

Pada proses pengujian juga terjadi *early stopping* pada epoch ke-32. Keadaan ini mengindikasikan bahwa model telah mencapai fase di mana kinerjanya pada dataset yang tidak digunakan dalam pelatihan utama, mulai tidak bergerak atau bahkan menurun, sehingga pelatihan dihentikan secara otomatis. Mekanisme ini krusial untuk mencegah *overfitting* dan menjamin model memiliki kemampuan generalisasi yang optimum terhadap data baru, serta meningkatkan efisiensi komputasi.



### 3. Hasil Pengembangan Pada Android

Pengujian aplikasi Android ini menerapkan Black Box testing. Metode ini berkonsentrasi pada validasi perilaku eksternal perangkat lunak. Penguji tidak perlu memahami struktur internal atau logika kode. Pendekatan ini sangat menekankan pada kepatuhan terhadap persyaratan aplikasi dan dilakukan dari sudut pandang pengguna (Praniffa et al., 2023). Tabel 2 menunjukkan hasil Black Box Testing dalam pengembangan aplikasi ini.

Tabel 2. Hasil Pengujian Black box

No.	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian	
			Berhasil	Tidak Berhasil
1	Menekan menu "Latihan"	Aplikasi beralih ke halaman menu Latihan yang menampilkan kamera	Ya	-
2	Menekan menu "Terjemahan"	Aplikasi beralih ke halaman menu Terjemahan yang menampilkan kolom inputan kata dan hasil terjemahan	Ya	-
3	Menekan menu "Kamus"	Aplikasi beralih ke halaman menu Kamus yang menampilkan beberapa kata	Ya	-
4	Menekan tombol enter saat menerjemahkan kata pada menu "Terjemahan"	Menampilkan gambar bahasa isyarat dari kata yang telah diinput	Ya	-
5	Menekan kata pada menu "Kamus"	Menampilkan video gerakan gambar bahasa isyarat dari kata yang dipilih	Ya	-

Hasil evaluasi fungsionalitas aplikasi yang ditampilkan dalam Tabel 2 menunjukkan bahwa semua lima skenario pengujian yang dilaksanakan mencapai status "Berhasil". Analisis ini menunjukkan bahwa setiap aspek yang diuji, mulai dari navigasi yang mulus saat menekan menu Latihan, Terjemahan, dan Kamus, berhasil menampilkan halaman dengan fitur kamera, kolom input terjemahan, dan daftar kata. Fungsi inti penerjemahan juga berfungsi dengan baik, di mana penekanan tombol enter setelah input kata berhasil menampilkan gambar bahasa isyarat. Selain itu, fitur kamus mampu memutar video gerakan bahasa isyarat saat sebuah kata dipilih, semuanya beroperasi sesuai dengan hasil yang diharapkan. Oleh karena itu, pengujian black box ini secara menyeluruh membuktikan bahwa aplikasi beroperasi dengan baik dan stabil dari sudut pandang pengguna.

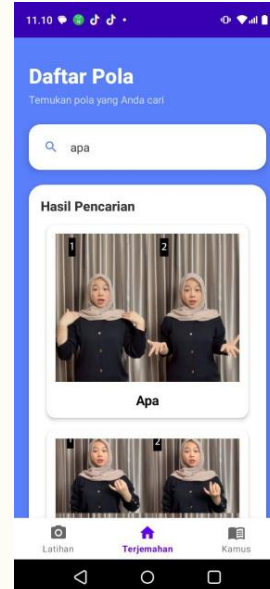
Pada pengembangan pada Android dilakukan dengan menggunakan Android Studio. Berikut merupakan tampilan UI dari aplikasi yang telah dibuat dan dikembangkan.



Gambar 11. Splash Screen



Gambar 12. Menu Latihan



Gambar 13. Menu Terjemahan



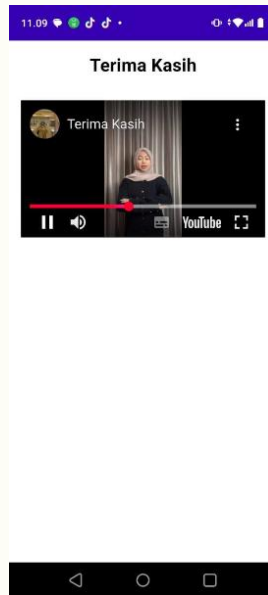
Gambar 14. Menu Kamus

Pada gambar 11. menampilkan halaman saat aplikasi pertama kali dibuka atau yang disebut sebagai splash screen. Pada halaman ini menampilkan nama aplikasi beserta dengan jargonnya.

Pada gambar 12. menampilkan antarmuka halaman Latihan, di mana pengguna memiliki kemampuan untuk memanfaatkan fitur ini dalam mengonversi gerakan tangan menjadi kata dalam bahasa isyarat. Pengguna dapat melakukan gerakan tangan yang membentuk bahasa isyarat, dan sistem akan menampilkan kata beserta tingkat akurasi. Gambar 12 menunjukkan bahwa kata "Tidur" memiliki akurasi 100%, karena bahasa isyarat untuk kata tersebut terdiri dari satu gerakan tunggal, berbeda dengan kata-kata lain yang melibatkan lebih dari satu gerakan. Sebelum pengguna dapat memanfaatkan fitur ini, akan muncul pop-up yang meminta izin untuk mengakses kamera perangkat dalam aplikasi.

Pada gambar 13. menampilkan menu Terjemahan. Halaman ini terdiri dari dua kolom, yaitu kolom pertama yang memungkinkan pengguna untuk memasukkan kata sementara kolom kedua menampilkan hasil gambar bahasa isyarat. Gambar bahasa isyarat yang ditampilkan pada menu Terjemahan tersimpan dalam MySQL. Ketika pengguna memasukkan sebuah kata, aplikasi akan mengirimkan kueri ke MySQL untuk memperoleh path atau URL dari gambar yang relevan. MySQL kemudian mengembalikan path tersebut, yang selanjutnya digunakan oleh aplikasi Android untuk memuat dan menampilkan gambar dari lokasi penyimpanannya.

Pada gambar 14. menampilkan menu Kamus yang berisi kata-kata bahasa isyarat yang telah diinput sebelumnya. Pengguna dapat memilih kata yang diinginkan, kemudian akan diarahkan ke halaman hasil video gerakan bahasa isyarat yang terintegrasi dengan YouTube, memungkinkan tampilan video gerakan isyarat secara efisien. Dengan memanfaatkan YouTube sebagai platform penyimpanan, aplikasi dapat menampilkan konten video dinamis tanpa secara signifikan membebani kapasitas penyimpanan perangkat. Gambar 15 menampilkan tampilan video gerakan bahasa isyarat.



Gambar 15. Hasil Menu Kamus

## KESIMPULAN

Penelitian ini berhasil menciptakan aplikasi Android penerjemah bahasa isyarat yang secara efektif mengintegrasikan pengenalan gerakan tangan. Implementasi arsitektur MobileNetV2, yang dilatih dengan dataset ekstensif, menghasilkan akurasi 76% dan total loss 0.81. Perlu ditekankan bahwa akurasi ini merupakan hasil awal (baseline) yang dicapai, mengindikasikan potensi yang signifikan namun masih terbuka lebar untuk peningkatan mengingat kompleksitas intrinsik pengenalan bahasa isyarat. Proses pelatihan dioptimalkan melalui penerapan mekanisme early stopping pada epoch ke-32, yang berfungsi sebagai strategi krusial untuk mencegah *overfitting* dan meningkatkan kemampuan generalisasi model. Aplikasi ini tidak hanya menyajikan fitur gerakan tangan atau Latihan, tetapi juga dilengkapi dengan fitur fungsional lainnya seperti fitur Terjemahan dan Kamus. Fitur Terjemahan dapat mengonversi teks menjadi gambar isyarat dengan dukungan MySQL sebagai katalog referensi. Dan pada fitur Kamus dapat menyajikan video gerakan isyarat terintegrasi dengan YouTube, menjadikan konektivitas internet esensial. Kontribusi ilmiah dari penelitian ini terletak pada pengembangan solusi holistik yang memanfaatkan *machine learning* untuk komunikasi, berfungsi sebagai alat bantu praktis dan sarana edukasi efektif bagi individu yang ingin berinteraksi atau mempelajari bahasa isyarat. Untuk pengembangan lebih lanjut, disarankan optimasi dan evaluasi mendalam terhadap model deteksi gerakan, termasuk pengujian pada dataset independen dan eksplorasi metode yang dapat menangani aspek temporal bahasa isyarat seperti LSTM atau 3D CNN, demi mencapai potensi maksimal aplikasi sebagai solusi yang lebih kuat dan menyeluruh.

## DAFTAR PUSTAKA

- Alfikri, R. H., Utomo, M. S., Februariyanti, H., & Nurwahyudi, E. (2022). Pembangunan Aplikasi Penerjemah Bahasa Isyarat Dengan Metode Cnn Berbasis Android. *Jurnal Teknoinfo*, 16(2), 183. <https://doi.org/10.33365/jti.v16i2.1752>
- Arisandi, L., & Satya, B. (2022). Sistem Klarifikasi Bahasa Isyarat Indonesia (Bisindo) Dengan Menggunakan Algoritma Convolutional Neural Network. *Jurnal Sistem Cerdas*, 5(3), 135-146. <https://doi.org/10.37396/jsc.v5i3.262>
- Bell, C. (2023). MySQL Database Service. *MySQL Database Service Revealed*, 137-195. [https://doi.org/10.1007/978-1-4842-8945-7\\_4](https://doi.org/10.1007/978-1-4842-8945-7_4)

- Borman, R. I., Priyopradono, B., & Syah, A. R. (2018). Klasifikasi Objek Kode Tangan pada Pengenalan Isyarat Alphabet Bahasa Isyarat Indonesia (BISINDO). *Seminar Nasional Informatika Dan Aplikasinya (SNIA), September*, 1–4. <https://doi.org/10.31227/OSF.IO/C7V2Z>
- Dewi, M., Wahyuningrum, T., & Prasetyo, N. A. (2021). Pengenalan Kata Bahasa Isyarat Indonesia (BISINDO) Menggunakan Augmented Reality (AR). *INISTA: Journal of Informatics, Information System, Software Engineering and Applications*, 3(2), 53–60. <https://journal.ittelkom-https://doi.org/10.20895/inista.v3i2.256>
- Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021). Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173(July 2020), 24–49. <https://doi.org/10.1016/j.isprsjprs.2020.12.010>
- Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, 3(1), 91–99. <https://doi.org/10.1016/j.gltp.2022.04.020>
- Nasha Hikmatia A.E., & Zul, M. I. (2021). Aplikasi Penerjemah Bahasa Isyarat Indonesia menjadi Suara berbasis Android menggunakan Tensorflow. *Jurnal Komputer Terapan*, 7(1), 74–83. <https://doi.org/10.35143/jkt.v7i1.4629>
- Nugraheni, A. S., Husain, A. P., & Unayah, H. (2023). Optimalisasi Penggunaan Bahasa Isyarat Dengan Sibi Dan Bisindo Pada Mahasiswa Difabel Tunarungu Di Prodi Pgmi Uin Sunan Kalijaga. *Jurnal Holistika*, 5(1), 28. <https://doi.org/10.24853/holistika.5.1.28-33>
- Nurhayati, O. D., Eridani, D., & Tsalavin, M. H. (2022). Sistem Isyarat Bahasa Indonesia (SIBI) Metode Convolutional Neural Network Sequential secara Real Time. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 9(4), 819–828. <https://doi.org/10.25126/jtiik.2022944787>
- Oliveira, V., Teixeira, L., & Ebert, F. (2020). On the Adoption of Kotlin on Android Development: A Triangulation Study. *SANER 2020 - Proceedings of the 2020 IEEE 27th International Conference on Software Analysis, Evolution, and Reengineering*, 206–216. <https://doi.org/10.1109/SANER48275.2020.9054859>
- Peran, M., Kecerdasan, P., Kualitas, B., & Hermawan, G. (2024). Memahami Peran Dataset dalam Penelitian Kecerdasan Buatan : Kualitas , Aksesibilitas , dan Tantangan. October. <https://doi.org/10.13140/RG.2.2.34468.49288>
- Praniffa, A., C., Syahri, A., Sandes, F., Fariha, U., Giansyah, Q., A., & Hamzah, M. (2023). Pengujian Sistem Informasi Parkir Berbasis Web Pada UIN SUSKA RIAU Menggunakan White Box dan Black Box Testing. *Jurnal Testing Dan Implementasi Sistem Informasi*, 1(1), 1–16. <https://doi.org/10.55583/jtisi.v1i1.321>
- Prasiwiningrum, E., & Lubis, A. (2024). Classification Of Palm Oil Maturity Using CNN ( Convolution Neural Network ) Modelling RestNet 50. *Decode* 4(3), 983–999. <https://doi.org/10.51454/decode.v4i3.822>
- Riyadhi, I. M., Intan Purnamasari, & Kamal Prihandani. (2023). Penerapan Pola Arsitektur Mvvm Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android. *INFOTECH Journal*, 9(1), 147–158. <https://doi.org/10.31949/infotech.v9i1.5246>
- Saputra, R., Wibowo, G. W. N., Zyen, A. K. (2025). Sistem Klasifikasi Alfabet Bahasa Isyarat Indonesia Menggunakan CNN dengan MobileNetV2 berbasis Android. *JUPITER* 17(1), 237–248. <https://doi.org/10.5281/zenodo.14686025>
- Sutarti, S., Syaqqialloh, F. (2025). Klasifikasi dan Pengenalan Emosi dari Ekspresi Wajah Menggunakan CNN-BiLSTM dengan Teknik Data Augmentation. *DECODE* 5(1), 79–91. <https://doi.org/10.51454/decode.v5i1.1038>

Wahyudi, J., Asbari, M., Sasono, I., Pramono, T., & Novitasari, D. (2022). Database Management Education in MYSQL. *Edumaspul: Jurnal Pendidikan*, 6(2), 2413–2417. <https://doi.org/10.33487/edumaspul.v6i2.4570>

Yunus, M., & Anwar, Y. (2022). Aplikasi Penerjemah Bahasa Isyarat Indonesia Ke Dalam Huruf Abjad. *Jurnal Sintaks Logika*, 2(1), 257–262. <https://doi.org/10.31850/jsilog.v2i1.1726>