

Pendeteksi Bahasa Isyarat Menggunakan TensorFlow dengan Metode *Convolutional Neural Network*

Reza Aditya Saputra^{1*}, Eddy Ryansyah¹, Fikri Maulana Setiawan¹, Chaerur Rozikin¹

¹ Fakultas Ilmu Komputer, Program Studi Informatika, Universitas Singaperbangsa Karawang, Karawang, Indonesia

Email: ^{1*}rezaadityasaputra50@gmail.com, ¹eddyryansyah1612@gmail.com, ¹maulamafikri@gmail.com,

¹chaerur.rozikin@staff.unsika.ac.id

Email Penulis Korespondensi: rezaadityasaputra50@gmail.com

Artikel Info :

Artikel History :

Submitted : 23-05-2025

Accepted : 26-05-2025

Published : 30-09-2025

Kata Kunci:

Pendeteksi; Bahasa

Isyarat; Pengelolaan

Citra; TensorFlow; CNN

Abstrak—Pengenalan bahasa isyarat memainkan peran penting dalam memfasilitasi komunikasi bagi individu dengan gangguan pendengaran. Penelitian ini mengusulkan model *Convolutional Neural Network* (CNN) yang dilatih untuk mengenali pola pada gambar bahasa isyarat dengan tujuan meningkatkan akurasi dan efisiensi sistem pengenalan bahasa isyarat. Model dilatih dalam dua tahap dengan sesi pelatihan pertama mencapai akurasi validasi sekitar 63%, sedangkan sesi pelatihan kedua menghasilkan akurasi validasi yang luar biasa melebihi 92% pada *epoch* ke-29. Peningkatan yang signifikan ini menunjukkan kemampuan model untuk belajar dan menggeneralisasi pola kompleks dalam gambar bahasa isyarat dengan efektif, menandakan potensinya untuk aplikasi praktis dalam interpretasi bahasa isyarat. Akurasi tinggi yang dicapai oleh model CNN menunjukkan kesesuaiannya untuk digunakan dalam berbagai skenario dunia nyata, seperti teknologi bantu untuk komunitas tunarungu atau sistem otomatisasi yang membutuhkan pengenalan gerakan tangan. Dengan demikian, model CNN yang dilatih ini memiliki potensi sebagai alat berharga dalam meningkatkan aksesibilitas dan efisiensi komunikasi bagi individu yang bergantung pada bahasa isyarat.

Abstract—Sign language recognition plays a vital role in facilitating communication for individuals with hearing impairments. This study proposes a *Convolutional Neural Network* (CNN) model trained to recognize patterns in sign language images with the aim of improving the accuracy and efficiency of sign language recognition systems. The model was trained in two stages with the first training session achieving a validation accuracy of around 63%, while the second training session yielded an impressive validation accuracy exceeding 92% at epoch 29. This significant improvement demonstrates the model's ability to effectively learn and generalize complex patterns in sign language images, signaling its potential for practical applications in sign language interpretation. The high accuracy achieved by the CNN model demonstrates its suitability for use in a variety of real-world scenarios, such as assistive technology for the deaf community or automation systems requiring hand gesture recognition. Thus, the trained CNN model has the potential to be a valuable tool in improving the accessibility and efficiency of communication for individuals who rely on sign language.

Keyword: Detector;

Sign Language; Image

Processing; TensorFlow;

CNN

1. PENDAHULUAN

Kemajuan teknologi digital telah memberikan dampak besar dalam berbagai aspek kehidupan manusia, termasuk dalam mendukung aksesibilitas komunikasi bagi individu penyandang disabilitas. Komunikasi adalah kebutuhan dasar manusia, namun bagi individu dengan gangguan pendengaran, komunikasi menjadi tantangan tersendiri karena keterbatasan dalam memahami maupun menyampaikan pesan secara verbal [1]. Bahasa isyarat menjadi media utama bagi komunitas tunarungu dalam berkomunikasi, namun tidak semua orang menguasai bahasa isyarat, sehingga masih terdapat hambatan dalam komunikasi dua arah antara penyandang disabilitas pendengaran dan masyarakat umum [2].

Dalam upaya menjembatani kesenjangan komunikasi tersebut, teknologi pengenalan bahasa isyarat berbasis citra mulai banyak dikembangkan. Sistem pengenalan bahasa isyarat bertujuan untuk menerjemahkan gestur tangan ke dalam teks atau suara, sehingga dapat dimengerti oleh orang awam. Teknologi ini umumnya memanfaatkan metode *computer vision* dan *machine learning*, khususnya pendekatan berbasis *deep learning* seperti *Convolutional Neural Network* (CNN) [3]. CNN merupakan salah satu algoritma dalam *deep learning* yang unggul dalam mengenali pola visual dari data citra dua dimensi, dan telah terbukti efektif dalam berbagai studi klasifikasi citra, deteksi objek, dan pengenalan gestur [4].

CNN bekerja dengan mengekstraksi fitur spasial dari citra melalui serangkaian lapisan konvolusi dan *pooling* yang kemudian diolah untuk menghasilkan prediksi terhadap kelas tertentu [5]. Arsitektur ini sangat cocok digunakan pada sistem yang memerlukan interpretasi visual dari citra, seperti pengenalan huruf bahasa isyarat yang memiliki bentuk tangan yang khas dan kompleks. TensorFlow sebagai salah satu kerangka kerja

pembelajaran mesin *open-source* menyediakan pustaka dan alat bantu yang memudahkan pembangunan, pelatihan, dan evaluasi model CNN secara efisien [6].

Beberapa penelitian sebelumnya telah mengembangkan sistem pengenalan bahasa isyarat dan sistem lainnya dengan berbagai pendekatan yang diterapkan dalam penelitian ini. Putra dan Saputra [7] membangun sistem deteksi penggunaan masker menggunakan CNN yang menunjukkan akurasi tinggi. Putra et al. [8] menerapkan TensorFlow dan CvZone dalam mendeteksi bahasa isyarat secara *real-time*, sedangkan Peling et al. [9] menggunakan TensorFlow Lite untuk menerapkan model ke perangkat seluler. Malik dan Zuliarso [10] menunjukkan penerapan CNN dalam deteksi jenis sayuran, sementara Gustsa dan Permadi [11] merancang sistem deteksi isyarat dengan TensorFlow *Object Detection*. Insani et al. [12] menggunakan pendekatan berbasis *Euclidean Distance*, namun akurasi masih tergolong rendah. Penelitian oleh Putri et al. [13] menggunakan metode LSTM untuk mengakomodasi urutan isyarat secara waktu nyata. Adapun studi oleh Robert et al. [14], serta Ardiansyah et al. [15] mengembangkan sistem pendeteksi bahasa isyarat menggunakan kamera *webcam* dengan metode *supervised learning* dan OpenCV.

Secara teknis, Python menjadi bahasa pemrograman pilihan utama dalam pengembangan sistem berbasis *machine learning* karena memiliki sintaks sederhana, pustaka lengkap, dan dukungan komunitas yang kuat [16]. Pengolahan citra digital pun menjadi tahap penting yang tidak terpisahkan dalam sistem pengenalan bahasa isyarat. Citra digital merupakan representasi dua dimensi dari objek dunia nyata yang disusun dari elemen piksel, dan pengolahan awal seperti normalisasi, augmentasi, konversi *grayscale* dan *resizing* menjadi tahap krusial dalam mempersiapkan data untuk proses pelatihan model [17].

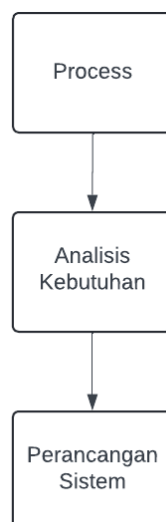
Meski berbagai pendekatan telah dilakukan, masih terdapat beberapa celah (gap) dalam penelitian sebelumnya, antara lain: (1) kurangnya eksperimen sistematis dalam melatih model CNN dengan tahapan bertingkat, (2) belum optimalnya pemanfaatan strategi seperti *early stopping callback* untuk efisiensi pelatihan, dan (3) belum banyak penelitian yang menguji performa CNN dalam konteks pengenalan huruf statis dari bahasa isyarat lokal Indonesia.

Berangkat dari permasalahan dan gap tersebut, penelitian ini bertujuan untuk merancang dan membangun sistem pendeteksi huruf bahasa isyarat berbasis CNN menggunakan TensorFlow dengan tahapan pelatihan bertingkat yang dievaluasi melalui nilai akurasi validasi model. Penelitian ini juga mengevaluasi efektivitas augmentasi citra dan strategi *callback* dalam meningkatkan generalisasi model. Diharapkan, sistem yang dibangun dapat menjadi solusi alternatif yang aplikatif dalam membantu komunikasi penyandang disabilitas melalui penerjemahan bahasa isyarat secara otomatis dan akurat, serta membuka peluang pengembangan ke arah pengenalan gestur yang lebih kompleks secara *real-time*.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilakukan melalui pendekatan eksperimen kuantitatif yang terstruktur, dimulai dari perencanaan sistem hingga evaluasi akhir dari model pengenalan huruf bahasa isyarat berbasis citra. Metodologi terdiri dari lima tahap utama: (1) persiapan dan pembentukan *dataset* citra bahasa isyarat, (2) augmentasi dan pra-proses data, (3) perancangan arsitektur model *Convolutional Neural Network* (CNN), (4) pelatihan model menggunakan TensorFlow, serta (5) evaluasi terhadap akurasi dan performa model. Seluruh proses dilakukan dalam lingkungan pemrograman Python dengan dukungan pustaka TensorFlow dan Keras.



Gambar 1. Alur proses *training model*

Berdasarkan alur proses pelatihan model yang ditunjukkan pada Gambar 1, maka tahapan dalam penelitian ini dapat dijelaskan sebagai berikut.

2.1.1 Process

Pada tahap ini dilakukan serangkaian proses awal dalam membangun sistem pengenalan huruf bahasa isyarat. Proses diawali dengan pengumpulan data berupa gambar tangan yang membentuk gestur bahasa isyarat. Gambar diambil menggunakan kamera *smartphone* dan dikelompokkan berdasarkan kelas gerakan tangan. Setelah itu, dilakukan augmentasi data seperti rotasi, *flipping*, *zooming*, dan *shifting* guna meningkatkan jumlah variasi data yang digunakan sebagai input pelatihan model. Selain augmentasi, dilakukan juga normalisasi citra dengan mengubah nilai piksel ke dalam rentang $[0, 1]$ agar proses pelatihan lebih stabil.

2.1.2 Analisis Kebutuhan

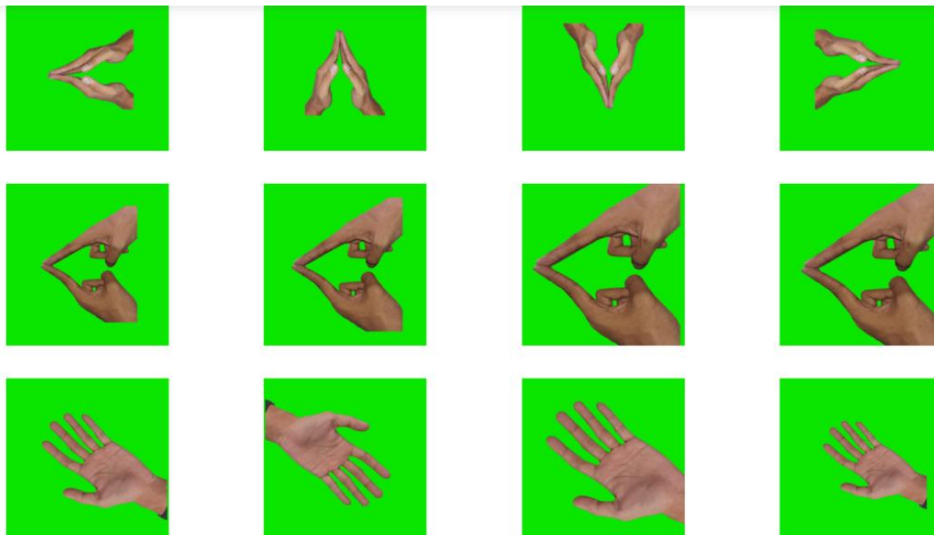
Tahap ini berfokus pada analisis kebutuhan sistem untuk menentukan pendekatan teknis dan alat yang akan digunakan. Analisis mencakup identifikasi kebutuhan perangkat keras dan lunak, pemilihan model *deep learning* (CNN), serta spesifikasi format input gambar dan jumlah kelas keluaran. Di tahap ini juga ditentukan bahwa sistem akan menggunakan bahasa pemrograman Python dengan pustaka TensorFlow dan Keras.

2.1.3 Perancangan Sistem

Pada tahap perancangan, peneliti membangun arsitektur model *Convolutional Neural Network* (CNN). Arsitektur terdiri atas tiga blok konvolusi yang masing-masing diikuti oleh *max pooling*, serta ditambahkan lapisan *dropout* untuk mencegah *overfitting*. Setelah itu, *output* di-*flatten*, dan diproses oleh lapisan *dense* dengan fungsi aktivasi *softmax* untuk klasifikasi. Desain sistem juga mencakup parameter pelatihan seperti jumlah *epoch*, *batch size*, dan penggunaan strategi *early stopping* sebagai mekanisme efisiensi waktu.

2.2 Persiapan Dataset

Tahap awal melibatkan pengumpulan data berupa gambar tangan yang membentuk huruf bahasa isyarat. Proses pengambilan gambar dilakukan secara langsung menggunakan kamera *smartphone* Xiaomi Redmi Note 8. Total gambar yang dikumpulkan adalah 1.512 citra, yang kemudian dibagi ke dalam dua kelompok: 1.210 gambar digunakan sebagai dataset pelatihan dan 302 gambar digunakan untuk validasi. Setiap gambar mewakili salah satu dari tiga kelas gerakan tangan: “Rumah”, “Tenda”, dan “Halo”. Untuk menjaga konsistensi dataset, semua gambar diambil dengan latar belakang yang seragam dan pencahayaan yang cukup. Gambar dikonversi ke ukuran dan format yang sesuai sebelum digunakan dalam pelatihan.



Gambar 2. Sampel gambar

2.3 Augmentasi dan Pra-proses Data

Untuk mengatasi keterbatasan jumlah data dan meningkatkan kemampuan generalisasi model, digunakan teknik augmentasi data melalui objek *ImageDataGenerator* dari pustaka Keras. Teknik ini bertujuan untuk menghasilkan variasi dari gambar asli melalui transformasi seperti rotasi, pergeseran posisi horizontal dan vertikal, *zoom*, serta *flipping* horizontal. Augmentasi ini juga bertujuan untuk mensimulasikan berbagai kondisi nyata yang mungkin dihadapi model saat digunakan. Selain augmentasi, dilakukan pula normalisasi

citra dengan me-*rescaling* nilai piksel gambar ke dalam rentang [0, 1]. Proses ini penting untuk mempercepat konvergensi selama pelatihan dan menjaga stabilitas numerik dari model.

2.4 Pembuatan Arsitektur Model

Model pengenalan bahasa isyarat yang digunakan dalam penelitian ini dirancang dengan arsitektur *Convolutional Neural Network* (CNN) yang sederhana namun efektif. Arsitektur terdiri dari tiga blok konvolusi yang masing-masing diikuti oleh operasi *max pooling*. Blok pertama menggunakan 16 filter konvolusi berukuran 3×3, blok kedua dan ketiga menggunakan 32 filter dengan ukuran kernel yang sama. Setelah tiga blok tersebut, model dilanjutkan dengan operasi *flatten*, yang bertugas mengubah *output* dari bentuk dua dimensi ke satu dimensi. Lapisan *dropout* dengan nilai 0.5 ditambahkan untuk mencegah *overfitting*. Akhirnya, lapisan *dense* digunakan dengan fungsi aktivasi *softmax* untuk melakukan klasifikasi ke tiga kelas berbeda. Arsitektur ini dipilih karena kemampuannya yang baik dalam mengekstraksi fitur spasial dari citra.

```
model = tf.keras.models.Sequential([  
  
    tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(150, 150, 3)),  
    tf.keras.layers.MaxPooling2D(2, 2),  
  
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),  
    tf.keras.layers.MaxPooling2D(2,2),  
  
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),  
    tf.keras.layers.MaxPooling2D(2,2),  
  
    tf.keras.layers.Flatten(),  
    tf.keras.layers.Dropout(0.5),  
  
    tf.keras.layers.Dense(128, activation='relu'),  
  
    tf.keras.layers.Dense(512, activation='relu'),  
  
    tf.keras.layers.Dense(3, activation='softmax')  
])
```

Gambar 3. Arsitektur model *Convolutional Neural Network* (CNN)

2.5 Pelatihan Gambar

Proses pelatihan gambar pada penelitian ini dilakukan dengan pendekatan berbasis *supervised learning* menggunakan arsitektur *Convolutional Neural Network* (CNN). Pelatihan dilakukan dalam dua sesi berbeda untuk mengeksplorasi peningkatan performa model melalui pengaturan parameter dan strategi pelatihan yang lebih terstruktur. Setiap sesi pelatihan direpresentasikan secara visual dalam bentuk grafik akurasi dan *loss* terhadap *epoch* pelatihan untuk mendukung pemahaman alur kerja model secara empiris. Gambar grafik pelatihan disisipkan sebagai bagian dari penjelasan.

```
Epoch 15/25
4/4 - 4s - loss: 1.0536 - accuracy: 0.4531 - val_loss: 1.0329 - val_accuracy: 0.5156
Epoch 16/25
4/4 - 4s - loss: 1.0468 - accuracy: 0.4531 - val_loss: 0.9630 - val_accuracy: 0.5703
Epoch 17/25
4/4 - 4s - loss: 1.0466 - accuracy: 0.5469 - val_loss: 1.0041 - val_accuracy: 0.5156
Epoch 18/25
4/4 - 5s - loss: 1.0163 - accuracy: 0.4297 - val_loss: 1.0410 - val_accuracy: 0.4766
Epoch 19/25
4/4 - 4s - loss: 1.0440 - accuracy: 0.4874 - val_loss: 1.0117 - val_accuracy: 0.5547
Epoch 20/25
4/4 - 4s - loss: 0.9956 - accuracy: 0.4922 - val_loss: 0.9243 - val_accuracy: 0.6641
Epoch 21/25
4/4 - 5s - loss: 0.8678 - accuracy: 0.6719 - val_loss: 1.3541 - val_accuracy: 0.3828
Epoch 22/25
4/4 - 5s - loss: 0.9192 - accuracy: 0.6328 - val_loss: 0.9101 - val_accuracy: 0.5781
Epoch 23/25
4/4 - 5s - loss: 0.9448 - accuracy: 0.5469 - val_loss: 0.8652 - val_accuracy: 0.6016
Epoch 24/25
4/4 - 5s - loss: 0.8286 - accuracy: 0.6797 - val_loss: 0.8485 - val_accuracy: 0.6250
Epoch 25/25
4/4 - 4s - loss: 0.8602 - accuracy: 0.6250 - val_loss: 0.7946 - val_accuracy: 0.6328
```

Gambar 4. Pelatihan pertama dari *dataset training* dan *validation*

Pada pelatihan pertama, model dilatih menggunakan metode `model.fit()` dengan total 25 *epoch* dan setiap *epoch* terdiri dari 4 langkah pelatihan (*steps per epoch*). Data pelatihan dan validasi dimuat menggunakan generator yang dibuat melalui `ImageDataGenerator`, dengan pembagian data sebesar 80% untuk pelatihan dan 20% untuk validasi. Tujuan dari pelatihan pertama adalah membangun *baseline* awal dari model CNN yang dirancang, serta melihat kecenderungan pola pembelajaran dari model terhadap dataset yang tersedia. Visualisasi grafik hasil pelatihan pertama dapat dilihat pada Gambar 4 yang menunjukkan tren akurasi dan *loss* dari proses pelatihan dan validasi.

```
16/16 - 12s - loss: 0.5094 - accuracy: 0.8105 - val_loss: 0.6231 - val_accuracy: 0.7266
Epoch 21/100
16/16 - 12s - loss: 0.5037 - accuracy: 0.8211 - val_loss: 0.4639 - val_accuracy: 0.8203
Epoch 22/100
16/16 - 12s - loss: 0.4974 - accuracy: 0.7930 - val_loss: 0.5203 - val_accuracy: 0.7891
Epoch 23/100
16/16 - 12s - loss: 0.5506 - accuracy: 0.7676 - val_loss: 0.5286 - val_accuracy: 0.7969
Epoch 24/100
16/16 - 12s - loss: 0.4751 - accuracy: 0.8203 - val_loss: 0.4507 - val_accuracy: 0.8438
Epoch 25/100
16/16 - 12s - loss: 0.5320 - accuracy: 0.8052 - val_loss: 0.4190 - val_accuracy: 0.8516
Epoch 26/100
16/16 - 12s - loss: 0.4337 - accuracy: 0.8330 - val_loss: 0.5295 - val_accuracy: 0.7656
Epoch 27/100
16/16 - 12s - loss: 0.3917 - accuracy: 0.8457 - val_loss: 0.3319 - val_accuracy: 0.8594
Epoch 28/100
16/16 - 12s - loss: 0.4320 - accuracy: 0.8429 - val_loss: 0.3507 - val_accuracy: 0.8906
Epoch 29/100
16/16 - 12s - loss: 0.4205 - accuracy: 0.8340 - val_loss: 0.2319 - val_accuracy: 0.9219
```

Gambar 5. Pelatihan kedua dari *dataset training* dan *validation*

Selanjutnya pada pelatihan kedua, konfigurasi pelatihan diperpanjang hingga 100 *epoch* dengan setiap *epoch* terdiri dari 16 langkah pelatihan. Perbedaan utama pada sesi pelatihan kedua adalah penggunaan strategi *early stopping* melalui metode *callback*. *Callback* ini dirancang untuk menghentikan pelatihan secara otomatis apabila akurasi validasi mencapai ambang batas $\geq 90\%$, dengan tujuan menghindari *overfitting* sekaligus meningkatkan efisiensi waktu pelatihan. Strategi ini memungkinkan proses pelatihan berhenti pada saat model telah mencapai performa optimal pada data validasi tanpa perlu melanjutkan hingga *epoch* ke-100.

3. HASIL DAN PEMBAHASAN

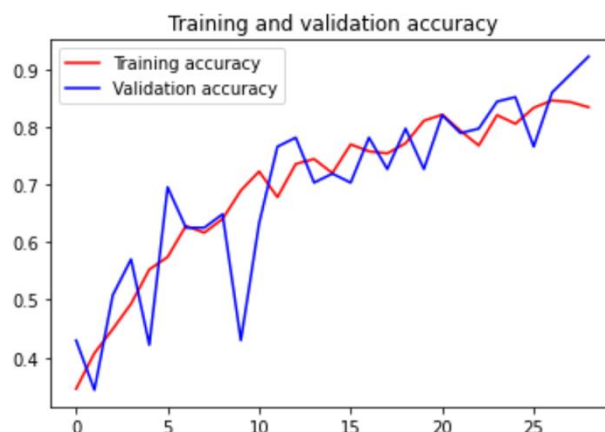
Penelitian ini menghasilkan model deteksi huruf bahasa isyarat berbasis citra digital menggunakan arsitektur *Convolutional Neural Network* (CNN) yang dibangun dan dilatih melalui pustaka TensorFlow. Proses pelatihan dilakukan dalam dua sesi, dan masing-masing sesi memberikan hasil performa yang berbeda. Tujuan utama dari proses pengujian ini adalah untuk mengevaluasi efektivitas model dalam mengenali pola dari citra tangan yang membentuk huruf dalam bahasa isyarat.



Gambar 6. Grafik pelatihan pertama *dataset*

Hasil pelatihan pertama menunjukkan bahwa model mencapai nilai akurasi validasi (*val_accuracy*) sebesar 0.6328. Metrik ini mencerminkan kemampuan awal model dalam melakukan klasifikasi terhadap citra validasi yang tidak pernah dilihat sebelumnya. Proses pelatihan selama 25 *epoch* memperlihatkan pola peningkatan performa secara bertahap, meskipun masih terdapat fluktuasi pada nilai *loss*, terutama karena keterbatasan data yang digunakan dan belum adanya penerapan strategi optimalisasi lanjutan seperti *callback*. Grafik akurasi dan *loss* dari pelatihan pertama ditampilkan pada Gambar 6.

Berdasarkan hasil tersebut, dilakukan pelatihan lanjutan untuk meningkatkan akurasi dan stabilitas model. Pada pelatihan kedua, model menggunakan konfigurasi *epoch* yang lebih banyak (hingga 100 *epoch*) serta menerapkan strategi *early stopping* berbasis *callback*. *Callback* diprogram untuk menghentikan pelatihan otomatis saat akurasi validasi mencapai 90% atau lebih. Hasil dari pelatihan kedua menunjukkan peningkatan performa yang signifikan, di mana nilai akurasi validasi mencapai 0.9219 dan pelatihan berhenti otomatis pada *epoch* ke-29. Hal ini menunjukkan bahwa model mampu belajar dan menggeneralisasi pola-pola visual dengan jauh lebih baik dibandingkan pelatihan pertama. Visualisasi grafik pelatihan kedua ditampilkan pada Gambar 7.



Gambar 7. Grafik pelatihan kedua *dataset*

Peningkatan performa dari pelatihan pertama ke pelatihan kedua mencerminkan bahwa strategi pelatihan yang digunakan memiliki dampak besar terhadap efisiensi pembelajaran model. Dengan penggunaan augmentasi data yang konsisten, penambahan *epoch*, dan *callback*, model tidak hanya memperoleh akurasi yang tinggi, tetapi juga menghindari gejala *overfitting*. Dari sudut pandang teknik CNN, lapisan-lapisan konvolusi yang digunakan telah berhasil mengekstrak fitur spasial dari gambar secara efektif, mulai dari tekstur, tepi, hingga bentuk tangan secara keseluruhan.

Untuk menguji kemampuan model dalam situasi nyata, dilakukan pengujian menggunakan citra baru yang tidak termasuk dalam *dataset* pelatihan maupun validasi. Model menunjukkan respons yang positif dengan melakukan prediksi kelas secara tepat pada sebagian besar gambar yang diberikan. Salah satu contoh hasil pengujian ditampilkan pada Gambar 8 yang menunjukkan klasifikasi citra input terhadap salah satu huruf bahasa isyarat.

test.png
Tangan Ini Menunjukkan Bahasa Isyarat Halo



Gambar 8. Pengujian dengan gambar

Keberhasilan model dalam mendeteksi citra baru menunjukkan bahwa jaringan CNN yang dilatih telah memiliki generalisasi yang baik terhadap pola-pola tangan. Keakuratan prediksi juga memperlihatkan ketepatan dalam pemetaan antara fitur visual dengan label kelas. Dengan demikian, model ini memiliki potensi tinggi untuk diimplementasikan dalam berbagai aplikasi dunia nyata seperti penerjemah bahasa isyarat otomatis berbasis perangkat lunak *desktop* atau aplikasi *mobile*.

Dari perspektif praktis, model ini dapat dimanfaatkan dalam bidang pendidikan inklusif, komunikasi penyandang disabilitas pendengaran, serta interaksi manusia dan mesin yang melibatkan gestur atau sinyal tangan. Dibandingkan dengan sistem tradisional berbasis sensor atau perangkat keras tambahan, sistem berbasis citra seperti ini memiliki keunggulan karena hanya membutuhkan kamera standar sebagai input. Hal ini tentunya memudahkan integrasi dalam platform yang lebih luas, seperti sistem informasi publik, layanan pelanggan, hingga media pembelajaran.

Namun demikian, beberapa tantangan tetap harus diperhatikan dalam pengembangan sistem sejenis. Salah satunya adalah sensitivitas terhadap pencahayaan dan latar belakang citra. Model yang dilatih dengan lingkungan pencahayaan seragam mungkin akan mengalami penurunan performa saat diuji pada kondisi pencahayaan yang berbeda. Oleh karena itu, pada tahap pengembangan lebih lanjut, disarankan untuk menambahkan data pelatihan dari berbagai kondisi pencahayaan dan latar yang lebih kompleks untuk memperkuat ketahanan model. Selain itu, klasifikasi lebih dari tiga kelas, seperti seluruh abjad atau gestur dinamis, memerlukan arsitektur yang lebih kompleks dan data yang jauh lebih besar.

Penelitian ini menunjukkan bahwa kombinasi antara CNN dan TensorFlow merupakan pendekatan yang efektif untuk sistem pengenalan citra bahasa isyarat. Dengan desain arsitektur yang tepat dan proses pelatihan yang optimal, sistem ini dapat mencapai tingkat akurasi tinggi yang cukup menjanjikan. Model yang dihasilkan dapat menjadi pondasi awal bagi pengembangan teknologi bantu komunikasi berbasis gestur di Indonesia.

Sebagai tambahan, keberhasilan ini juga membuka potensi integrasi dengan teknologi *real-time detection* atau dengan kerangka kerja tambahan seperti MediaPipe atau OpenCV untuk aplikasi interaktif yang lebih canggih. Penerapan pada perangkat bergerak seperti *smartphone* juga memungkinkan karena TensorFlow menyediakan dukungan untuk *deployment* menggunakan TensorFlow Lite.

Dengan hasil pelatihan yang telah dicapai, maka dapat disimpulkan bahwa sistem pendeteksi bahasa isyarat berbasis CNN ini telah memenuhi ekspektasi awal dalam mengenali gestur tangan secara akurat dan efisien. Seluruh hasil dan temuan akan dirangkum dan ditekankan dalam bagian kesimpulan.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan metode *Convolutional Neural Network* (CNN) yang dibangun menggunakan kerangka kerja TensorFlow mampu memberikan hasil yang signifikan dalam proses pengenalan huruf bahasa isyarat. Model yang dikembangkan menunjukkan peningkatan akurasi yang

signifikan antara sesi pelatihan pertama dan kedua, yaitu dari 63% menjadi 92% pada *epoch* ke-29 setelah penerapan strategi *callback* untuk menghentikan pelatihan saat akurasi validasi telah mencapai ambang batas tertentu. Hasil ini mengindikasikan bahwa arsitektur model yang dirancang telah mampu menangkap fitur-fitur visual kompleks dalam citra bahasa isyarat dan melakukan generalisasi terhadap data yang belum pernah dilihat sebelumnya secara efektif.

Efektivitas model dalam mengenali pola isyarat huruf didukung oleh tahapan pra-proses seperti augmentasi data melalui ImageDataGenerator dan pemilihan struktur arsitektur CNN yang terdiri atas beberapa lapisan konvolusi, *pooling*, dan *dropout*. Model ini berpotensi besar untuk diimplementasikan dalam sistem bantu komunikasi bagi penyandang disabilitas pendengaran, serta dalam pengembangan sistem pengenalan gestur otomatis untuk kebutuhan lain yang relevan dalam bidang interaksi manusia-komputer (*human-computer interaction*).

Dengan mempertimbangkan hasil yang diperoleh, dapat disimpulkan bahwa penggunaan CNN dalam pendeteksian bahasa isyarat memberikan kontribusi nyata terhadap kemajuan teknologi inklusif. Penelitian ini juga membuka peluang untuk pengembangan lebih lanjut, seperti pengenalan kata atau kalimat bahasa isyarat secara *real-time* serta integrasi dengan perangkat keras berbasis *embedded system*. Diharapkan penelitian ini dapat menjadi fondasi dalam pengembangan sistem pengenalan bahasa isyarat yang lebih komprehensif dan akurat di masa mendatang.

REFERENCES

- [1] M. A. Imaddudin, I. W. Hamzah, and S. Astuti, "Simulasi Penerjemah SIBI (Sistem Isyarat Bahasa Indonesia) Menggunakan Tensorflow Dan Convolutional Neural Network (CNN)," *e-Proceeding Eng.*, vol. 8, no. 6, pp. 3911–3918, 2022, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/19146>
- [2] N. T. Adam, Z. A. Tyas, and T. Hardiani, "Deteksi Gestur Sistem Isyarat Bahasa Indonesia Menggunakan Metode Deep learning SSD MobileNet V2 FPNLite," *Sainteks*, vol. 21, no. 2, p. 129, Oct. 2024, doi: 10.30595/sainteks.v21i2.24006.
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
- [4] J. Anggara, E. Ryansyah, and B. Arif Dermawan, "IMPLEMENTASI OBJECT DETECTION DALAM KLASIFIKASI SAMPAH UNTUK MENINGKATKAN EFISIENSI PENGELOLAAN LIMBAH," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 9, no. 3, pp. 4923–4930, May 2025, doi: 10.36040/jati.v9i3.13813.
- [5] A. Rohim, Y. A. Sari, and T. Tibyani, "Convolution Neural Network (CNN) Untuk Pengklasifikasian Citra Makanan Tradisional," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 7, pp. 7038–7042, 2019, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5851>
- [6] Nasha Hikmatia A.E. and M. I. Zul, "Aplikasi Penerjemah Bahasa Isyarat Indonesia menjadi Suara berbasis Android menggunakan Tensorflow," *J. Komput. Terap.*, vol. 7, no. 1, pp. 74–83, Jun. 2021, doi: 10.35143/jkt.v7i1.4629.
- [7] D. R. R. Putra and R. A. Saputra, "IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK MENDETEKSI PENGGUNAAN MASKER PADA GAMBAR," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3, pp. 710–714, Aug. 2023, doi: 10.23960/jitet.v11i3.3286.
- [8] I. N. T. A. Putra, K. S. Kartini, Y. K. Suyitno, I. M. Sugiarta, and N. K. E. Puspita, "Penerapan Library Tensorflow, Cvzone, dan Numpy pada Sistem Deteksi Bahasa Isyarat Secara Real Time," *J. Krisnadana*, vol. 2, no. 3, pp. 412–423, May 2023, doi: 10.58982/krisnadana.v2i3.335.
- [9] I. B. A. Peling, I. M. P. A. Ariawan, and G. B. Subiksa, "Deteksi Bahasa Isyarat Menggunakan Tensorflow Lite dan American Sign Language (ASL)," *J. Krisnadana*, vol. 3, no. 2, pp. 90–100, Jan. 2024, doi: 10.58982/krisnadana.v3i2.534.
- [10] R. A. Malik and E. Zuliarso, "Metode Convolutional Neural Network Untuk Mendeteksi Jenis Sayur Menggunakan Tensorflow," *Media Bina Ilm.*, vol. 15, no. 12, pp. 5873–5882, 2021, [Online]. Available: <https://ejournal.binawakya.or.id/index.php/MBI/article/view/1147>
- [11] A. H. Gustsa and G. S. Permadi, "Sistem Deteksi Bahasa Isyarat Secara Realtime Dengan Tensorflow Object Detection dan Python Menggunakan Metode Convolutional Neural Network," *Inov. J. Ilm. Inov. ...*, vol. 7, no. 2, pp. 1–10, 2023, [Online]. Available: <https://ejournal.unhasy.ac.id/index.php/inovate/article/view/4116>
- [12] C. N. Insani, N. Arifin, and M. R. Rasyid, "Deteksi Gerakan Bahasa Isyarat Menggunakan Euclidean Distance," *Inform. J. Ilmu Komput.*, vol. 19, no. 1, pp. 99–106, May 2023, doi: 10.52958/iftk.v19i1.5658.
- [13] H. M. Putri, F. Fadlisyah, and W. Fuadi, "PENDETEKSIAN BAHASA ISYARAT INDONESIA SECARA REAL-TIME MENGGUNAKAN LONG SHORT-TERM MEMORY (LSTM)," *J. Teknol. Terap. Sains 4.0*, vol. 3, no. 1, p. 663, Mar. 2022, doi: 10.29103/tts.v3i1.6853.

- [14] D. Robert, M. Nababan, and Z. Budiarso, "Sistem Pendeteksi Gerakan Bahasa Isyarat Indonesia Menggunakan Webcam Dengan Metode Supervised Learning," *J. Ilm. Komputasi*, vol. 22, no. 3, pp. 449–456, Oct. 2023, doi: 10.32409/jikstik.22.3.3403.
- [15] A. R. Ardiansyah, A. H. Nur'azizan, and R. Fernandis, "Implementasi Deteksi Bahasa Isyarat Tangan Menggunakan OpenCV dan MediaPipe," *Stain. (Seminar Nas. Teknol. Sains)*, vol. 3, no. 1, pp. 331–337, 2024, [Online]. Available: <https://proceeding.unpkediri.ac.id/index.php/stains/article/view/4337>
- [16] F. Chollet, *Deep Learning with Python*, Second Edi. Shelter Island: Manning Publications Co., 2021. [Online]. Available: <https://www.manning.com/books/deep-learning-with-python-second-edition>
- [17] D. P. Mawardi, M. Novita, and N. Dwi Saputro, "Deteksi Awal Klasifikasi Jenis Penyakit Kanker Kulit Dengan Algoritma Convolutional Neural Network (CNN) Berbasis Mobile Apps," *Adopsi Teknol. dan Sist. Inf.*, vol. 3, no. 2, pp. 1–6, Dec. 2024, doi: 10.30872/atasi.v3i2.2305.