

PROGRAM PENERJEMAH BAHASA ISYARAT INDONESIA (BISINDO) SECARA *REAL TIME* MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK* DAN *MEDIAPIPE*

Renaldy William Lijaya Therry, Achmad Junaidi , Andreas Nugroho Sihananto
Teknik Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur
Jalan Raya Rungkut Madya No. 1, Gunung Anyar, Surabaya, Indonesia
renaldytherry225@gmail.com

ABSTRAK

Komunikasi adalah salah satu aspek penting dalam kehidupan sehari-hari, namun bagi komunitas tuna rungu dan tuna wicara, komunikasi dapat menjadi tantangan besar. Bahasa isyarat Indonesia (BISINDO) merupakan salah satu alat komunikasi yang digunakan oleh komunitas ini, namun masih banyak masyarakat umum yang tidak memahami bahasa isyarat tersebut. Hal ini dapat menyebabkan kesulitan dalam berkomunikasi dan membatasi akses komunitas tuna rungu dan tuna wicara dalam berpartisipasi dalam kegiatan sosial dan ekonomi. Oleh karena itu, pengembangan sistem deteksi bahasa isyarat yang akurat sangat penting untuk memfasilitasi komunikasi antara komunitas tuna rungu, tuna wicara dan masyarakat umum. Pada penelitian ini, penulis mengembangkan program penerjemah bahasa isyarat Indonesia menggunakan *Convolutional Neural Network* (CNN) dan *MediaPipe*, program ini memiliki keterbatasan, yaitu arah pembacaan proses deteksi berada pada tangan kanan dan program ini hanya dapat mendeteksi abjad BISINDO. Hasil pelatihan model menunjukkan akurasi sebesar 98.59%, *loss* sebesar 13.03%, akurasi validasi sebesar 93.78%, *validation loss* sebesar 32.15%. Hasil evaluasi model, model dapat mendeteksi abjad bahasa isyarat Indonesia dengan akurasi sebesar 94%,... Hasil pengujian akurasi deteksi pada empat skenario Hasil deteksi terbaik diperoleh pada skenario pertama, yaitu pencahayaan baik dan latar belakang gelap dengan nilai akurasi sebesar 72%, presisi sebesar 69.5%, recall sebesar 68% dan F1-score sebesar 68.7%.

Kata kunci : CNN, *Mediapipe*, Bisindo.

1. PENDAHULUAN

Bahasa isyarat adalah metode komunikasi non-verbal yang banyak digunakan oleh individu dengan gangguan pendengaran atau bicara. Namun, kurangnya pemahaman dari orang-orang yang tidak menguasai bahasa isyarat dapat menyebabkan kesulitan dalam berinteraksi dan berintegrasi dengan masyarakat luas. Hal ini dapat berdampak pada kualitas hidup individu dengan gangguan pendengaran atau bicara, seperti kesulitan dalam mengakses pendidikan, pekerjaan, dan layanan kesehatan. Selain itu, kesulitan dalam berkomunikasi juga dapat menyebabkan perasaan isolasi dan kesepian pada individu dengan gangguan pendengaran atau bicara. [1]

Bahasa Isyarat Indonesia (BISINDO) adalah bahasa isyarat yang paling umum digunakan oleh komunitas tuna rungu dan tuna wicara. Namun, masih banyak masyarakat umum yang tidak memahami bahasa isyarat ini [2], sehingga diperlukan sebuah program yang dapat menerjemahkan bahasa isyarat ke dalam bahasa lisan atau tulisan secara otomatis dan akurat.

Teknologi visi komputer dan kecerdasan buatan dapat dimanfaatkan untuk mengembangkan sistem deteksi bahasa isyarat. Dengan menggunakan kamera sebagai sensor, sistem dapat mengenali gerakan tangan pengguna bahasa isyarat dan mengklasifikasikannya ke dalam huruf, angka, atau kata yang sesuai. Salah satu metode yang populer digunakan dalam bidang visi komputer dan kecerdasan buatan adalah Jaringan Saraf

Konvolusional (CNN), sebuah jenis jaringan saraf tiruan yang terdiri dari beberapa lapisan konvolusi, aktivasi, pooling, dan sepenuhnya terhubung. CNN dapat belajar secara otomatis fitur penting dari gambar input dan menghasilkan output dalam bentuk kelas atau label. [2]

Terdapat beberapa metode yang digunakan untuk mendeteksi bahasa isyarat, yaitu, YOLO [3], LSTM dan mediapipe [4], Faster R-CNN [5], Tensorflow *Object detection API* [6], dll. Pada penelitian ini metode yang digunakan adalah CNN, karena CNN memiliki beberapa kelebihan di antaranya adalah dapat mempelajari pola dari data citra mentah, memiliki akurasi yang tinggi, dapat mempelajari pola spasial secara hierarki (high level dan both level) sehingga dapat memproses gambar yang kompleks [4], [7]. Metode tambahan yang digunakan adalah Mediapipe, sebuah kerangka kerja lintas platform untuk membangun pipeline pembelajaran mesin terapan multimodal (persepsi dengan visi, pendengaran, dan lainnya). Mediapipe dapat memberikan alat untuk mendeteksi dan mendeskripsikan fitur lokal dari gambar, menjadi invarian terhadap transformasi geometris dan pencahayaan sehingga meningkatkan akurasi dan kecepatan deteksi. Pada penelitian ini mediapipe digunakan untuk mengekstraksi dan memproses pola tangan pada gambar untuk diperoleh koordinat yang akan membantu deteksi bahasa isyarat [8]–[10]. Dengan menggabungkan Mediapipe dan CNN, diharapkan dapat menghasilkan program penerjemah bahasa isyarat yang efektif dan efisien.

2. TINJAUAN PUSTAKA

2.1. Bahasa Isyarat Indonesia

Bahasa isyarat merupakan bentuk komunikasi yang berbeda dari bahasa lisan, yang menekankan pada aspek suara dan verbal. Orang tuli menggunakan bahasa isyarat dengan aspek visual, menggunakan tangan, bahu, mata, alis, dan ekspresi wajah lainnya untuk berkomunikasi. Karena perbedaan ini, orang tuli seringkali menghadapi tantangan dalam berinteraksi dalam masyarakat, dengan tingkat keterampilan literasi rata-rata setara dengan anak berusia 10 tahun. Meski mereka memiliki kemampuan visual, namun kemampuan ini tidak sebanding dengan orang yang dapat mendengar yang belajar bahasa tulis sebagai representasi visual dari bahasa lisan. [11]

Bisindo, juga dikenal sebagai Bahasa Isyarat Indonesia, adalah salah satu cara berkomunikasi dalam bahasa isyarat di Indonesia, selain metode lainnya seperti SIBI atau Sistem Isyarat Bahasa Indonesia. Biasanya, Bisindo digunakan oleh komunitas tuli dalam percakapan sehari-hari mereka. Meskipun orang tuli sering kali menghadapi tantangan dalam berinteraksi dengan yang lain, Bisindo menawarkan petunjuk konseptual, yang sangat berbeda dari Kamus SIBI yang berfokus pada petunjuk struktural. Tetapi kedua bentuk petunjuk ini dapat digunakan dan dikembangkan oleh penggunanya, karena keduanya memiliki tujuan yang sama, yaitu untuk meningkatkan pendidikan, martabat, dan kualitas hidup serta sumber daya manusia bagi orang tuli di Indonesia. Salah satu tantangan yang dihadapi oleh komunitas tuli adalah bagaimana cara menginformasikan pembelajaran bahasa isyarat yang dapat dimengerti oleh orang yang tidak tuli, sehingga keduanya dapat berkomunikasi, menjalin pertemanan, dan mendukung kehidupan mereka. [12]

2.2. Convolutional Neural Network

CNN atau Convolutional Neural Network adalah jenis model deep learning yang digunakan untuk memproses data dengan pola grid, seperti gambar. Model ini terinspirasi dari organisasi korteks visual pada hewan dan dirancang untuk belajar secara otomatis dan adaptif fitur-fitur hierarkis spasial, mulai dari pola-pola sederhana hingga kompleks. [13]

CNN terdiri dari tiga jenis lapisan yang merupakan blok pembangunannya. Lapisan konvolusi dan lapisan pooling bertanggung jawab dalam mengambil fitur-fitur dari data, sedangkan lapisan terhubung sepenuhnya digunakan untuk menghubungkan fitur-fitur tersebut dengan output akhir, seperti klasifikasi. [13]

Lapisan konvolusi dalam CNN mengekstrak fitur penting dari data input, seperti gambar, dengan menerapkan filter (kernel) yang bergerak di atas data. Setiap operasi konvolusi menghitung produk dot antara bobot filter dan data input, menghasilkan peta fitur yang mewakili fitur-fitur seperti tepi, bentuk, dan tekstur. Peta fitur ini digunakan sebagai input untuk lapisan berikutnya dalam jaringan, memungkinkan

CNN menangkap fitur lokal dan mempertahankan hubungan spasial antar piksel untuk tugas pengenalan gambar atau klasifikasi. [14]

Banyak arsitektur CNN telah dikembangkan selama beberapa tahun, masing-masing dengan fitur unik dan aplikasinya. Beberapa arsitektur yang paling terkenal termasuk LeNet-5, AlexNet, VGGNet, dan ResNet. CNNs memiliki rentang aplikasi yang luas karena kemampuan mereka untuk menangkap pola dalam gambar. Beberapa contoh aplikasinya yaitu pengenalan (klasifikasi) gambar, analisis gambar medis, pemrosesan bahasa alami, dan analisis video. Mereka digunakan dalam pengenalan wajah, deteksi objek pada mobil self-driving, tugas klasifikasi gambar umum, dan mendiagnosa penyakit dengan menganalisis gambar medis. [11], [15]

2.3. Mediapipe

Mediapipe adalah kerangka kerja yang memungkinkan pengembang untuk membangun *pipeline* ML terapan lintas platform yang multi-modal (video, audio, data seri waktu apa pun). MediaPipe memiliki koleksi besar model deteksi dan pelacakan tubuh manusia yang dilatih pada dataset Google yang sangat besar dan paling beragam. Sebagai kerangka simpul dan tepi atau landmarks, mereka melacak titik kunci di berbagai bagian tubuh. Semua titik koordinat dinormalisasi dalam tiga dimensi. [13].

21 titik kunci 3D yang mewakili tangan dapat diubah menjadi koordinat relatif dengan memilih pusat telapak tangan (atau pergelangan tangan) sebagai titik referensi. Kemudian untuk setiap titik kunci, kurangi koordinat pusat telapak tangan dari koordinat titik kunci dengan menggunakan persamaan. Normalisasi koordinat relatif dengan mengikuti langkah-langkah yang dijelaskan pada paragraf berikutnya, kemudian ratakan koordinat yang dinormalisasi menjadi vektor 2D. Dengan menggunakan koordinat relatif bukan koordinat absolut, kita dapat membuat hand gesture recognition lebih tahan terhadap perubahan ukuran dan orientasi tangan. [10]

2.4. Confusion Matrix

Matriks kebingungan (confusion matrix) adalah metode untuk mengukur kinerja deteksi objek seperti akurasi, presisi, dan recall. Akurasi diperoleh berdasarkan rasio prediksi yang benar terhadap data keseluruhan. Presisi diperoleh berdasarkan rasio prediksi positif yang benar dibandingkan dengan hasil prediksi positif keseluruhan. Perlu diingat bahwa itu berbasis rasio prediksi positif yang benar dibandingkan dengan data yang benar keseluruhan. Dalam matriks kebingungan, terdapat empat istilah atau empat keputusan yang mungkin terjadi dalam mendeteksi objek, seperti *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). Perhitungan akurasi, presisi, dan recall menggunakan persamaan matriks kebingungan yang ditampilkan pada Gambar 1 dengan nilai akurasi,

presisi, dan recall yang ditampilkan pada Persamaan 1, persamaan 2, dan persamaan 3. [4], [6]

		Actual Class	
		Positive (P)	Negative (N)
Predicted Class	Positive (P)	True Positive (TP)	False Positive (FP)
	Negative (N)	False Negative (FN)	True Negative (TN)

Gambar 1. Confusion Matrix

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

3. METODE PENELITIAN

Dalam melaksanakan penelitian ini terdapat beberapa tahap yang dilaksanakan yaitu studi literatur, pengumpulan data, pemrosesan data, pelatihan model, evaluasi model, dan pengujian.

3.1. Studi Literatur

Pada awal penelitian, penulis melakukan studi literatur dengan tujuan untuk menemukan tahapan-tahapan dalam membuat program penerjemah bahasa isyarat indonesia menggunakan convolutional neural network dan mediapipe menggunakan referensi dari jurnal.

3.2. Pengumpulan Data

Dataset yang digunakan dalam penelitian ini terdiri dari gambar pose tangan sesuai abjad bahasa isyarat indonesia, yang dikumpulkan dari berbagai sumber untuk memastikan keberagaman data. Dataset ini dikumpulkan dan dikompilasi dari sumber-sumber seperti Kaggle, Roboflow, dan foto-foto yang diambil oleh peneliti sendiri. Total dataset yang terkumpul sebanyak 16828 gambar, yang kemudian dibagi menjadi tiga subset, yaitu set pelatihan sebanyak 11788 gambar, set pengujian sebanyak 2520 gambar, dan set validasi sebanyak 2520 gambar. Selanjutnya gambar tersebut dipisahkan ke masing-masing kelas abjad A-Z.

3.3. Pemrosesan Data

Pada awal tahap pemrosesan data, gambar-gambar pada dataset di-load, selanjutnya gambar dikonversi ke RGB; kemudian proses pendeteksi hand landmark dilakukan dengan menggunakan mediapipe hands solution, pada proses ini mediapipe mendeteksi

titik-titik koordinat tangan, kemudian terjadi proses pembuatan bounding box yang mencakup kedua tangan, kemudian gambar dipotong sampai menyisakan tangan yang berada pada bounding box. Selanjutnya, gambar-gambar pada dataset di ubah ukurannya ke 128x128, dan di konversi dari BGR ke grayscale. Selanjutnya, label A-Z dikonversi ke numerik menggunakan one hot encoder (A = 0, B = 1, C = 2, dst).

3.4. Pelatihan Model

Pada tahap pelatihan model, gambar-gambar yang telah diproses sebelumnya dilatih dengan arsitektur Convolutional Neural Network (CNN) yang dirancang untuk mengenali pola-pola pada gambar dan membuat prediksi yang akurat. Model ini menggunakan beberapa lapisan convolutional untuk mengekstrak fitur-fitur dari gambar, diikuti oleh lapisan batch *normalization*, *pooling*, *flatten*, dan *dense* untuk membuat prediksi. Parameter yang digunakan termasuk jumlah filter, kernel size, strides, padding, dan activation function, serta penggunaan lapisan dropout untuk mencegah overfitting. Model ini dikompilasi dengan optimizer Adam, serta diuji dengan metrics akurasi. Dengan demikian, proses pelatihan model ini dapat membantu meningkatkan kinerja model dalam mengenali pola-pola pada gambar.

3.5. Evaluasi Model

Setelah pelatihan model selesai, evaluasi model dilakukan dengan menggunakan dua metode utama, yaitu classification report dan confusion matrix. Classification report menampilkan hasil precision, recall, f1-score, dan support untuk mengevaluasi kinerja model dalam memprediksi kelas-kelas yang berbeda. Precision, recall, dan f1-score memberikan informasi tentang akurasi model dalam memprediksi kelas-kelas, sedangkan support menunjukkan jumlah contoh data yang digunakan. Sementara itu, confusion matrix menampilkan hasil prediksi model dalam bentuk matriks, memperlihatkan seberapa banyak hasil prediksi yang benar dan salah untuk setiap kelas. Dengan demikian, kedua metode ini dapat membantu mengevaluasi kinerja model dalam memprediksi kelas-kelas yang berbeda.

3.6. Pengujian

Tahapan selanjutnya adalah tahap pengujian, pengujian dilakukan pada 4 skenario untuk mengukur akurasi deteksi, skenario pertama adalah tempat dengan pencahayaan bagus dan latar belakang berwarna cenderung gelap, skenario kedua adalah tempat dengan pencahayaan bagus dan latar belakang berwarna cerah, skenario ketiga adalah tempat dengan pencahayaan buruk dan latar belakang berwarna cerah dan skenario keempat adalah tempat dengan pencahayaan buruk dan latar cenderung gelap.

4. HASIL DAN PEMBAHASAN

4.1. Pemrosesan Data

Pada awal tahap pemrosesan data, gambar-gambar pada dataset di-load ke dalam memori komputer. Proses ini dilakukan untuk mempersiapkan data yang akan diproses oleh sistem. Setelah gambar-gambar di-load, maka dilakukan proses konversi warna gambar dari format BGR (*Blue, Green, Red*) ke format RGB (*Red, Green, Blue*). Proses konversi ini dilakukan karena beberapa algoritma pengolahan gambar memerlukan format RGB.

Selanjutnya, proses pendeteksi *hand landmark* dilakukan dengan menggunakan *MediaPipe Hands Solution*. *MediaPipe* adalah sebuah library yang dikembangkan oleh Google yang dapat digunakan untuk melakukan pengolahan gambar dan video. Pada proses ini, *MediaPipe* mendeteksi titik-titik koordinat tangan pada gambar, sehingga dapat dibuat sebuah *bounding box* yang mencakup kedua tangan. *Bounding box* adalah sebuah kotak yang dibuat untuk mengelilingi objek yang ingin diproses.

Setelah *bounding box* dibuat, maka gambar dipotong sampai menyisakan hanya tangan yang berada pada *bounding box*. Proses ini dilakukan untuk menghilangkan bagian gambar yang tidak relevan dengan objek yang ingin diproses.

Selanjutnya, gambar-gambar pada dataset diubah ukurannya ke 128x128 piksel. Proses ini dilakukan untuk memastikan bahwa semua gambar memiliki ukuran yang sama, sehingga dapat diproses oleh sistem dengan lebih efisien. Setelah itu, gambar-gambar dikonversi dari format BGR ke grayscale. Proses ini dilakukan karena beberapa algoritma pengolahan gambar lebih mudah diimplementasikan pada gambar grayscale.

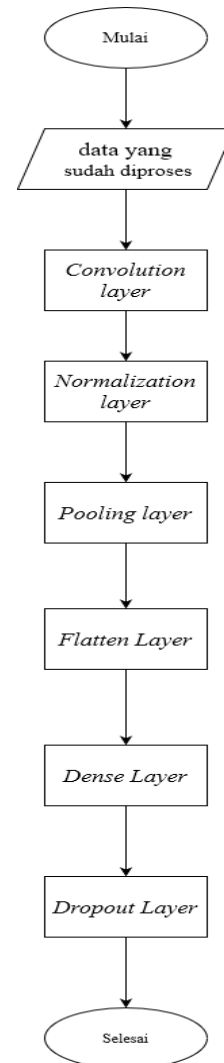
Terakhir, label A-Z dikonversi ke numerik menggunakan *one hot encoder*. *One hot encoder* adalah sebuah teknik yang digunakan untuk mengubah label kategori menjadi bentuk numerik. Pada kasus ini, label A-Z dikonversi menjadi numerik dengan menggunakan aturan sebagai berikut: A = 0, B = 1, C = 2, dan seterusnya. Proses ini dilakukan untuk memastikan bahwa label dapat diproses dengan lebih efisien.

4.2. Pelatihan Model

Tahap pelatihan model. Pada tahap ini, gambar-gambar yang telah diproses sebelumnya mulai dilatih dengan arsitektur *Convolutional Neural Network* (CNN) yang telah dibuat. CNN ini dirancang untuk dapat mengenali pola-pola pada gambar dan membuat prediksi yang akurat.

Pada awalnya, model dilatih di lapisan *convolutional*, lapisan ini berfungsi untuk mendeteksi fitur-fitur lokal pada gambar, seperti garis, sudut, dan bentuk. Lapisan *convolutional* ini menggunakan teknik konvolusi untuk mendeteksi pola-pola pada gambar, sehingga dapat mengenali fitur-fitur yang relevan dengan tugas yang diinginkan.

Selain itu, *kernel initializer* digunakan untuk menginisialisasi bobot kernel, yaitu parameter yang digunakan dalam proses konvolusi. *Kernel initializer* ini berfungsi untuk memberikan nilai awal pada bobot kernel, sehingga model dapat memulai proses pembelajaran dengan baik.



Gambar 2. Flowchart Convolutional Neural Network

Kernel regularizer juga digunakan untuk mengatur besar bobot kernel, sehingga model tidak terlalu kompleks dan dapat menghindari *overfitting*. Dengan menggunakan *kernel regularizer*, model dapat mempertahankan kemampuan generalisasi yang baik dan menghindari *overfitting* pada data pelatihan.

Dengan demikian, model dapat dilatih untuk mendeteksi fitur-fitur pada gambar, seperti garis, sudut, dan bentuk, dengan menggunakan lapisan *convolutional* dan *kernel initializer*. Selain itu, *kernel regularizer* dapat digunakan untuk mengatur besar bobot kernel dan menghindari *overfitting*.

Setelah lapisan *convolutional*, model dilatih dengan lapisan *batch normalization*. Lapisan *batch normalization* berfungsi untuk menormalkan output dari lapisan *convolutional* sebelumnya, sehingga output memiliki distribusi yang lebih stabil dan dapat

membantu meningkatkan kinerja model. Dengan demikian, lapisan batch normalization dapat membantu meningkatkan kinerja model dan mengurangi overfitting.

Kemudian, model dilatih dengan lapisan pooling yang berfungsi untuk mengurangi dimensi fitur-fitur yang diekstrak sebelumnya. Lapisan pooling yang digunakan adalah *MaxPool2D*. Lapisan pooling ini dapat membantu mengurangi dimensi fitur-fitur dan meningkatkan kinerja model.

Setelah itu, model dilatih dengan lapisan flatten. Lapisan flatten bekerja dengan cara mengubah output dari lapisan convolutional dan batch normalization yang berbentuk tiga dimensi menjadi vektor satu dimensi. Hal ini dilakukan dengan cara mengubah setiap elemen dari output menjadi satu vektor panjang yang dapat diolah oleh lapisan densitas berikutnya.

Setelah itu, model dilatih dengan lapisan dense yang berperan sebagai lapisan output. Lapisan dense bekerja dengan cara menghubungkan setiap neuron pada lapisan sebelumnya dengan setiap neuron pada lapisan dense. Hal ini memungkinkan lapisan dense untuk memproses informasi yang lebih kompleks dan membuat prediksi yang lebih akurat. Selain itu, lapisan dense juga dapat meningkatkan kinerja model dengan cara mengurangi kesalahan prediksi dan meningkatkan kecepatan pelatihan model.

Learning rate Adam optimizer yang digunakan dalam model ini adalah sebesar 0.0001. Learning rate ini digunakan untuk mengatur kecepatan pembelajaran model, sehingga model dapat belajar dengan lebih efektif dan efisien. Dengan menggunakan learning rate sebesar 0.0001, model dapat belajar dengan lebih stabil dan tidak terlalu cepat atau terlalu lambat. Hal ini dapat membantu model untuk menghindari overfitting dan underfitting, sehingga model dapat membuat prediksi yang lebih akurat.

Terakhir, model dilatih pada lapisan dropout dengan tujuan meningkatkan kemampuan generalisasi dan mencegah overfitting. Dropout adalah teknik yang digunakan untuk mengurangi kompleksitas model dengan menonaktifkan (mengatur menjadi nol) sejumlah neuron secara acak selama pelatihan. Dengan menggunakan dropout, model dipaksa untuk belajar menggunakan subset neuron yang berbeda-beda selama pelatihan, sehingga meningkatkan kemampuan generalisasi dan mencegah overfitting.

Proses pelatihan model dengan dropout melibatkan beberapa langkah. Pertama, model dilatih pada sebagian besar data pelatihan. Kemudian, sebagian neuron dalam model dilakukan dropout secara acak, sehingga neuron tersebut tidak berkontribusi pada hasil prediksi. Model dipaksa untuk belajar menggunakan neuron lainnya untuk mencapai performa yang sama. Model dilatih secara bergantian dengan menggunakan neuron yang berbeda-beda, sehingga meningkatkan kemampuan generalisasi. Proses ini diulangi beberapa kali hingga model mencapai performa yang optimal.

Dengan menggunakan dropout, model dapat meningkatkan kemampuan generalisasi dan mengurangi risiko overfitting, sehingga meningkatkan performa pada data baru yang belum pernah dilihat. Teknik ini sangat efektif dalam meningkatkan kemampuan model untuk memprediksi hasil yang akurat pada data yang belum pernah dilihat sebelumnya.

Berikut ini adalah hasil pelatihan model

Tabel 1. Hasil Pelatihan Model

<i>Epoch</i>	31
<i>loss</i>	13.03%
<i>Accuracy</i>	98.59%
<i>Validation loss</i>	32.15%
<i>Validation accuracy</i>	93.78%

Dari hasil pelatihan model pada tabel 1 dapat diketahui bahwa akurasi yang diraih dari hasil pelatihan model sebesar 98.59%, *loss* sebesar 13.03%, akurasi validasi sebesar 93.78%, *validation loss* sebesar 32.15%. Hasil ini menunjukkan bahwa dalam proses pelatihan model dapat mendeteksi abjad bahasa isyarat Indonesia dengan baik.

4.3. Evaluasi Model

Setelah pelatihan model selesai, evaluasi model dilakukan dengan menggunakan dua metode utama, yaitu classification report dan confusion matrix. Kedua metode ini digunakan untuk mengevaluasi kinerja model dalam memprediksi kelas-kelas yang berbeda.

Classification report menampilkan hasil precision, recall, f1-score, dan support pada setiap epoch pelatihan model. Precision adalah ukuran yang menunjukkan seberapa banyak hasil prediksi yang benar dari total hasil prediksi yang dilakukan. Recall adalah ukuran yang menunjukkan seberapa banyak hasil prediksi yang benar dari total hasil prediksi yang seharusnya dilakukan. F1-score adalah ukuran yang menunjukkan rata-rata dari precision dan recall. Support adalah ukuran yang menunjukkan jumlah contoh data yang digunakan untuk menghitung precision, recall, dan f1-score. Dengan menggunakan classification report, dapat diketahui seberapa baik model dalam memprediksi kelas-kelas yang berbeda.

Confusion matrix menampilkan hasil prediksi model dalam bentuk matriks, di mana sumbu x menunjukkan hasil prediksi yang benar (true labels) dan sumbu y menunjukkan hasil prediksi yang salah (predicted labels). Dengan demikian, dapat diketahui seberapa banyak hasil prediksi yang benar dan salah untuk setiap kelas.

Tabel 2. *Classification Report* Model pada set *testing*

<i>Precision</i>	0.95
<i>Recall</i>	0.94
<i>F1-sccore</i>	0.94
<i>Accuracy</i>	0.94

Perhitungan *Classification Report* pada tabel 2 berasal dari persamaan berikut

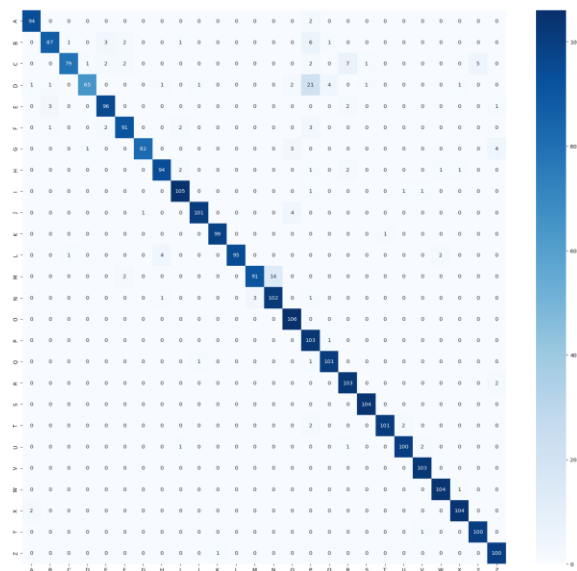
$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

Dari hasil pengujian model pada tabel 2 menunjukkan hasil performa model pada set *testing* dengan berbagai metrik seperti *precision*, *recall*, *f1-score*, dan *support* untuk masing-masing kelas. Dari laporan ini, terlihat bahwa model memiliki akurasi keseluruhan sebesar 0.94 yang menunjukkan bahwa 94% dari prediksi yang dibuat oleh model pada set pengujian adalah benar.

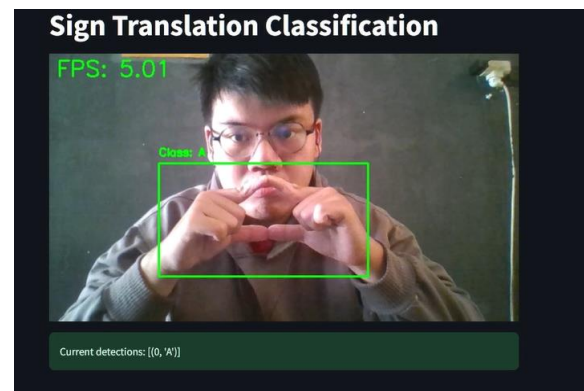


Gambar 3. *Confusion matrix* set *testing*

Gambar 2 adalah *confusion matrix* dari set pengujian (*testing*) model deteksi Bahasa Isyarat Indonesia (Bisindo) menggunakan CNN (*Convolutional Neural Network*) dan *MediaPipe*. *Confusion matrix* adalah matriks yang menggambarkan performa model klasifikasi dengan membandingkan prediksi model terhadap label sebenarnya.

Dari matriks ini, dapat ditarik kesimpulan bahwa model memiliki kemampuan klasifikasi yang cukup baik pada data pengujian. Sebagian besar nilai-nilai di diagonal utama (yang menunjukkan jumlah prediksi yang benar untuk masing-masing kelas) memiliki nilai yang cukup tinggi, mengindikasikan bahwa model dapat mengklasifikasikan kelas-kelas dengan akurasi yang baik.

4.4. Implementasi Program



Gambar 4. Tampilan Program Penerjemah Bisindo

Model yang sudah dibuat, dijalankan dengan program, program ini dibuat dengan menggunakan bahasa pemrograman *python* dan *framework Streamlit* gambar 3 adalah tampilan program. Alur berjalannya program adalah sebagai berikut, pengguna membuka program, program mengakses *webcam* laptop, pengguna menampilkan pose tangan, dan selanjutnya program melakukan proses deteksi pose tangan dan menampilkan hasil deteksi abjad bisindo.



Gambar 5. *Flowchart* program

4.5. Pengujian

Tahap pengujian dilakukan pada 4 skenario untuk mengukur akurasi deteksi, skenario pertama adalah tempat dengan pencahayaan bagus dan latar belakang berwarna cenderung gelap, skenario kedua adalah tempat dengan pencahayaan bagus dan latar belakang berwarna cerah, skenario ketiga adalah tempat dengan pencahayaan buruk dan latar belakang berwarna cerah dan skenario keempat adalah tempat dengan pencahayaan buruk dan latar cenderung gelap. Berikut ini adalah hasil deteksi abjad bisindo pada keempat skenario.

Tabel 3. *Classification Report* Skenario Pengujian

Skenario	Akurasi	Presisi	Recall	F1-score
1	0.720	0.695	0.680	0.687
2	0.550	0.520	0.540	0.529
3	0.389	0.410	0.375	0.391
4	0.320	0.298	0.315	0.306

Perhitungan *Classification Report* pada tabel 2 berasal dari persamaan berikut

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

Hasil deteksi terbaik diperoleh pada skenario pertama, yaitu pencahayaan baik dan latar belakang gelap dengan nilai akurasi sebesar 72%, presisi sebesar 69.5%, *recall* sebesar 68% dan F1-score sebesar 68.7%. Hal ini disebabkan karena kontras antara objek dan latar belakang lebih baik, membuatnya lebih mudah dideteksi oleh CNN. Pencahayaan yang baik juga mengurangi noise pada gambar, sehingga memudahkan deteksi. Oleh karena itu, skenario pertama merupakan skenario ideal untuk deteksi objek menggunakan CNN. Selain itu, kualitas kamera perangkat juga mempengaruhi hasil pengujian skenario.

5. KESIMPULAN DAN SARAN

Penelitian ini mengembangkan program penerjemah bahasa isyarat indonesia (Bisindo_ menggunakan algoritma Convolutional Neural Network (CNN) dan Mediapipe. Program ini dibangun menggunakan bahasa pemrograman Python dan framework *streamlit*, namun program ini memiliki keterbatasan, yaitu arah pembacaan proses deteksi hanya berada pada tangan kanan dan program ini hanya dapat mendeteksi abjad BISINDO. Hasil pelatihan model menunjukkan akurasi sebesar

98.59%, loss sebesar 13.03%, akurasi validasi sebesar 93.78%, *validation loss* sebesar 32.15%. Hasil evaluasi model, model dapat mendeteksi abjad bahasa isyarat Indonesia dengan akurasi sebesar 94%. Hasil pengujian akurasi deteksi pada empat skenario Hasil deteksi terbaik diperoleh pada skenario pertama, yaitu pencahayaan baik dan latar belakang gelap dengan nilai akurasi sebesar 72%, presisi sebesar 69.5%, *recall* sebesar 68% dan F1-score sebesar 68.7%. Untuk meningkatkan performa model lebih lanjut, disarankan untuk meningkatkan jumlah dataset yang lebih besar dan beragam, sehingga model dapat lebih baik dalam mendeteksi bahasa isyarat.

DAFTAR PUSTAKA

- [1] S. Bae, "A Study on the Difficulty of Communication through Sign Language in Non-English Speaking Countries," *Open J. Soc. Sci.*, vol. 11, no. 10, pp. 494–506, 2023, doi: 10.4236/jss.2023.1110028.
- [2] N. A. Hasma, F. Arnia, R. Muharar, and M. K. Muchamad, "Pengenalan Gerakan Isyarat Bahasa Indonesia Menggunakan Algoritma Surf Dan K-Nearest Neighbor," *J. Komputer, Inf. Teknol. dan Elektro*, vol. 7, no. 1, p. 2022, 2022, [Online]. Available: <https://jurnal.unsyiah.ac.id/kitektro/article/view/23262>
- [3] S. Daniels, N. Suciati, and C. Fathichah, "Indonesian Sign Language Recognition using YOLO Method," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1077, no. 1, p. 012029, 2021, doi: 10.1088/1757-899x/1077/1/012029.
- [4] B. Sundar and T. Bagyammal, "American Sign Language Recognition for Alphabets Using MediaPipe and LSTM," *Procedia Comput. Sci.*, vol. 215, pp. 642–651, 2022, doi: 10.1016/j.procs.2022.12.066.
- [5] R. A. Alawwad, O. Bchir, and M. M. Ben Ismail, "Arabic Sign Language Recognition using Faster R-CNN," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 3, pp. 692–700, 2021, doi: 10.14569/IJACSA.2021.0120380.
- [6] L. N. Hayati, A. N. Handayani, W. S. G. Irianto, R. A. Asmara, D. Indra, and M. Fahmi, "Classifying BISINDO Alphabet using TensorFlow Object Detection API," *Ilk. J. Ilm.*, vol. 15, no. 2, pp. 358–364, 2023, [Online]. Available: <https://jurnal.fikom.umi.ac.id/index.php/ILKOM/article/view/1692>
- [7] J. Ren, Y. Wang, J. Ren, and Y. Wang, "Overview of Object Detection Algorithms Using Convolutional Neural Networks," *J. Comput. Commun.*, vol. 10, no. 1, pp. 115–132, 2022, doi: 10.4236/jcc.2022.101006.
- [8] M. Z. Fauzi and R. Sarno, "Recognition of Real-Time Angklung Kodály Hand Gesture using Mediapipe and Machine Learning Method," *ICCoSITE 2023 - Int. Conf. Comput. Sci. Inf.*

- Technol. Eng. Digit. Transform. Strateg. Facing VUCA TUNA Era*, pp. 980–985, 2023, doi: 10.1109/ICCoSITE57641.2023.10127808.
- [9] J. S. Han, C. I. Lee, Y. H. Youn, and S. J. Kim, “A Study on Real-time Hand Gesture Recognition Technology by Machine Learning-based MediaPipe,” *J. Syst. Manag. Sci.*, vol. 12, no. 2, pp. 468–482, 2022, doi: 10.33168/JSMS.2022.0225.
- [10] V. Shinde, T. Bacchav, J. Pawar, and M. Sanap, “Hand Gesture Recognition System Using Camera,” *B.E Comput. Eng.*, pp. 1–8, 2020, [Online]. Available: www.ijert.org
- [11] T. Handhika, R. I. M. Zen, Murni, D. P. Lestari, and I. Sari, “Gesture recognition for Indonesian Sign Language (BISINDO),” *J. Phys. Conf. Ser.*, vol. 1028, no. 1, 2018, doi: 10.1088/1742-6596/1028/1/012173.
- [12] U. Fadlilah, D. Wismoyohadi, A. K. Mahamad, and B. Handaga, “Bisindo information system as potential daily sign language learning,” *AIP Conf. Proc.*, vol. 2114, 2019, doi: 10.1063/1.5112492.
- [13] A. Halder and A. Tayade, “Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning,” *Int. J. Res. Publ. Rev.*, vol. 2, no. 5, pp. 9–17, 2021, [Online]. Available: www.ijrpr.com
- [14] M. Kumar, P. Gupta, R. K. Jha, A. Bhatia, K. Jha, and B. K. Shah, “Sign language alphabet recognition using convolution neural network,” *Proc. - 5th Int. Conf. Intell. Comput. Control Syst. ICICCS 2021*, no. Iccics, pp. 1859–1865, 2021, doi: 10.1109/ICICCS51141.2021.9432296.
- [15] R. Rastgoo, K. Kiani, and S. Escalera, “Video-based isolated hand sign language recognition using a deep cascaded model,” *Multimed. Tools Appl.*, vol. 79, no. 31–32, pp. 22965–22987, 2020, doi: 10.1007/s11042-020-09048-5.