Sistem Pendeteksian Bahasa Isyarat Alfabet Sibi Menggunakan Algoritma *You Only Look Once* Dan *Roboflow*

1st Jean Jeasen Timotius
Fakultas Teknik Elektro
S1 Teknik Komputer
Bandung, Indonesia
jeanjeasen@student.telkomuniversity.ac.id

2nd Casi Setianingsih
Fakultas Teknik Elektro
S1 Teknik Komputer
Bandung, Indonesia
setiacasie@telkomuniversity.ac.id

3rd Marisa W. Paryasto
Fakultas Teknik Elektro
S1 Teknik Komputer
Bandung, Indonesia
marisaparyasto@telkomuniversity.ac.id

Bahasa isyarat merupakan bahasa yang digunakan oleh penyandang tunarungu untuk berkomunikasi dengan orang lain. Minimnya akses informasi dan komunikasi bagi penyandang disabilitas, khususnya tuna rungu, mendorong dikembangkannya solusi yang mampu menerjemahkan bahasa isyarat ke dalam bentuk teks dan ucapan. Melihat permasalahan dan celah tersebut, penelitian ini dilakukan untuk mengembangkan sebuah sistem penerjemahan bahasa isyarat alfabet SIBI dengan menggunakan algoritma YOLOv8. Algoritma YOLOv8 digunakan untuk mengenali dan menerjemahkan karakter bahasa SIBI ke dalam teks bahasa Indonesia. Dataset yang digunakan terdiri dari gambar yang diambil secara manual serta bersumber dari internet. Hasil pengujian menunjukkan bahwa model yang dilatih dengan dataset split 80: 10: 10 mencapai akurasi tertinggi, dengan nilai mAP50 sebesar 0,99 dan nilai mAP50-95 sebesar 0,91. Selain itu, optimizer SGD memberikan kinerja terbaik dibandingkan dengan optimizer lain, seperti Adam dan RMSprop. Sistem penerjemahan bahasa isyarat sebagai produk dalam penelitian ini diharapkan dapat meningkatkan aksesibilitas penyandang tunarungu serta pembelajaran bahasa isyarat di Indonesia.

Kata kunci: Bahasa isyarat, machine learning, optimizer, SIBI, YOLOv8

Sign language is a language used by deaf people to communicate with others. The lack of access to information and communication for people with disabilities, especially the deaf, encourages the development of solutions that are able to translate sign language into text and speech form. Seeing these problems and gaps, this research was conducted to develop a SIBI alphabet sign language translation system using the YOLOv8 algorithm. The YOLOv8 algorithm is used to recognize and translate SIBI language characters into Indonesian text. The dataset used consists of images taken manually as well as those sourced from the internet. The test results show that the model trained with 80:10:10 split dataset achieves the highest accuracy, with mAP50 value of 0.99 and mAP50-95 value of 0.91. Besides that, the SGD optimizer provides the best performance compared to other optimizers such as Adam and RMSprop. The sign language translation system as a product in this research is expected to improve the accessibility of deaf people and sign language learning in Indonesia.

Keywords: machine learning, optimizer, sign language, SIBI, YOLOv8

I. PENDAHULUAN

Masalah utama yang dihadapi penyandang disabilitas, khususnya tunarungu, adalah kurangnya aksesibilitas terhadap informasi dan komunikasi. Penggunaan bahasa isyarat sebagai cara utama untuk berkomunikasi akan lebih mudah diakses bagi penyandang tunarungu apabila bahasa tersebut dapat diterjemahkan ke dalam bentuk verbal atau nonverbal. Hal ini menunjukkan pentingnya penyediaan medium bagi komunitas ini untuk akses terhadap informasi yang lebih baik.

Bahasa isyarat adalah bahasa yang yang disampaikan lewat penggunaan bahasa tubuh, gerak bibir, bentuk tangan, serta gerak tangan, dan wajah untuk mengungkapkan pikirannya [1]. Bahasa ini digunakan oleh penyandang tunarungu sebagai alat komunikasi utamanya. Di berbagai belahan dunia, terdapat jenis bahasa isyarat yang berbeda, misalnya *American Sign Language* (ASL) di Amerika Serikat dan Sistem Isyarat Bahasa Indonesia (SIBI) di Indonesia. Meskipun digunakan sebagai alat komunikasi utama oleh penyandang tunarungu, bahasa isyarat masih sulit dipahami bahkan digunakan oleh masyarakat umum non penyandang disabilitas.

Pembelajaran bahasa isyarat umumnya dipelajari melalui media cetak seperti buku pembelajaran atau kamus bahasa isyarat yang diterbitkan oleh Kementerian Pendidikan dan Kebudayaan Indonesia. Tidak hanya lewat buku, pembelajaran bahasa isyarat juga dapat diakses dengan menyewa jasa seorang penerjemah bahasa isyarat. Namun, satu kali sesi pembelajaran bahasa isyarat memerlukan biaya sekitar Rp1.500.000, sehingga praktiknya masih minim dilakukan akibat keterbatasan biaya [2]. Oleh karena itu, dibutuhkan sebuah solusi untuk mengatasi masalah pembelajaran bahasa isyarat yang awalnya hanya terpaku pada metode konvensional dengan

kebutuhan biaya yang cukup besar, menjadi sebuah metode pembelajaran yang interaktif, minim biaya, dan dapat digunakan dengan mudah baik bagi penyandang tunarungu maupun masyarakat secara umum.

Penelitian yang mengkaji bahasa isyarat sudah banyak dilakukan oleh berbagai pihak dalam bahasa yang beragam, mulai dari penggunaan sensor untuk mendeteksi bahasa isyarat hingga proses penerjemahannya menggunakan bantuan *machine learning*. Namun, penelitian yang mengkaji penerjemahan bahasa isyarat masih sedikit dilakukan, terutama di Indonesia. Hal ini dikarenakan proses penerjemahan di Indonesia mayoritas masih dilakukan secara konvensional melalui media buku sehingga proses penerjemahan memakan waktu yang cukup lama.

Pada penelitian ini, *machine learning* digunakan sebagai alat bantu untuk mempelajari bahasa isyarat Indonesia (SIBI) dan menerjemahkannya ke dalam bahasa Indonesia. Adanya sebuah sistem yang dapat menerjemahkan bahasa isyarat menjadi bahasa indonesia diharapkan dapat mempercepat proses pembelajaran bahasa isyarat. Selain itu, pembelajaran yang dilakukan menggunakan bantuan sebuah sistem juga memiliki tingkat akurasi yang akurat, sehingga tiap gerakan dalam bahasa isyarat akan diterjemahkan dengan benar. Penelitian ini dilakukan untuk melihat apakah penerjemahan bahasa isyarat dengan bantuan *machine learning* dapat dilakukan untuk menerjemahkan bahasa isyarat Indonesia terutama pada bahasa isyarat SIBI.

II. KAJIAN TEORI

Dengan perkembangan teknologi yang semakin pesat, terutama pada bidang komputer, banyak penelitian dilakuakn untuk mengembangkan sebuah sistem komputer yang dapat mengerjakan pekerjaan manusia, yaitu *machine learning* [3], [4]. Sistem *machine learning* ini dapat digunakan sebagai alat bantu dalam proses pembelajaran dan penerjemahan bahasa isyarat Indonesia yang cepat dan akurat. Pada bahasa isyarat indonesia SIBI, terdapat 4 kelas utama yaitu: angka, huruf, kata, dan imbuhan [5]. Setiap kelas ini memiliki gerakan dan arti yang berbeda-beda, sehingga *machine learning* digunakan untuk menerjemahkan bahasa isyarat menjadi bahasa Indonesia. Penelitian ini berfokus untuk meneliti penerjemahan bahasa isyarat SIBI pada kelas huruf, yang terdiri dari 26 kelas, yaitu huruf A-Z. Terdapat beberapa penelitian yang sebelumnya menggunakan berbagai algoritma *machine* learning untuk melakukan penerjemahan bahasa isyarat. Beberapa diantaranya seperti CNN (*Convolutional Neural Networks*), LSTM (*Long Short-Term Memory*), dan YOLO (*You Only Look Once*). Penelitian serupa juga pernah dilakukan oleh Elek Alaftek, Ishak Pacal, dan Kenan Cicek dari *Research Institute of Neural Computing and Applications.* Penelitian ini menggunakan YOLOv4-CSP untuk menerjemahkan bahasa isyarat Turki. Model YOLOv4-CSP dilatih dengan *dataset* berlabel yang terdiri dari angka-angka dalam bahasa isyarat Turki, dan perbandingan kinerja mereka dalam mengenali isyarat tangan. Metode yang diusulkan menghasilkan presisi 98,95%, *recall* 98,15%, 98,55% skor F1 dan 99,49%.[6]

III. METODE

Penelitian ini menggunakan algoritma YOLO (*You Only Look Once*) untuk melakukan penerjemahan bahasa isyarat SIBI dengan metode *image to text translate.* Algoritma YOLO dikembangkan oleh Ultralytics dan digunakan untuk berbagai macam tujuan, termasuk deteksi objek, klasifikasi gambar, dan lain-lain [7]. Dalam penelitian ini, algoritma YOLO digunakan untuk deteksi objek pada gambar alfabet SIBI, yang terdiri dari 26 kelas, khususnya huruf A-Z. Penelitian ini juga memanfaatkan Roboflow untuk memudahkan distribusi data, perbandingan, pelabelan data, dan penyimpanan *dataset.*

Pembuatan algoritma YOLO untuk penerjemahan bahasa isyarat SIBI perlu melewati beberapa langkah teknis yang penting. Langkah tersebut termasuk pengambilan kumpulan data, pemrosesan awal kumpulan data, pelabelan kumpulan data, dan pelatihan model algoritme. Berikut adalah tahapan- tahapan yang dilakukan untuk membuat model YOLO untuk menerjemahkan bahasa isyarat SIBI.

A. Pengumpulan *Dataset*

Algoritme YOLOv8 secara eksklusif menggunakan kumpulan data gambar. Alfabet SIBI harus dikonversi ke dalam format gambar untuk dapat diklasifikasikan tanpa hambatan oleh algoritma YOLOv8. Sangat penting untuk memastikan bahwa *dataset* gambar mencakup beragam variasi alfabet SIBI, mengingat ada 26 kelas alfabet SIBI yang berbeda. Kelas-kelas ini meliputi huruf A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, dan Z. Kumpulan data terdiri dari gambar-gambar yang diambil secara manual dan bersumber dari Kaggle, dengan informasi terperinci tentang sumber gambar yang disertakan dalam setiap kumpulan data yang disediakan pada bagian selanjutnya.

TABEL 1.
DETAIL INFORMASI GAMBAR UNTUK *DATASET*.

No	Metode Pengambilan	*Dataset*	Total *Dataset*
1	Dari Internet (Kaggle)	*Hand Sign Language Detection*	5200
2	Manual	Mahasiswa Telkom University	2600

Berikut ini dilampirkan gambar dari beberapa dataset alfabet SIBI.

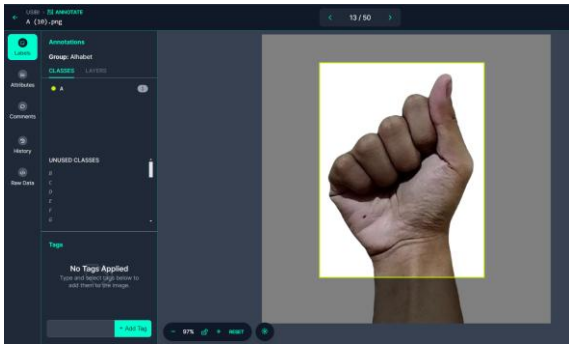
TABEL 2.
CONTOH *DATASET* PADA ALFABET SIBI.

Huruf	Gambar
A	

B	
C	

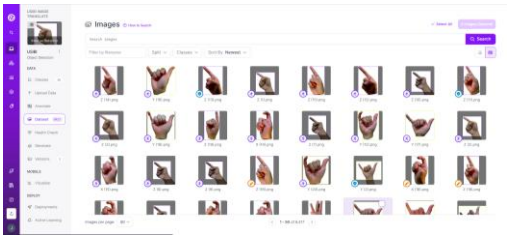
B. *Preprocessing Dataset*

Pada *prepossessing dataset*, setiap gambar *dataset* harus berukuran 640x640 piksel. Jika data tidak memenuhi persyaratan ini, ukurannya dapat diubah secara manual atau menggunakan fungsi Python untuk mempercepat proses. Gambar dengan ukuran yang benar dapat diunggah ke Roboflow, di mana gambar tersebut akan diberi anotasi untuk menunjukkan kelas alfabet SIBI yang benar. Berikut ini adalah garis besar proses *labelling* kumpulan data gambar di Roboflow



GAMBAR 1.
PROSES *LABELLING DATASET* PADA ROBOFLOW

Untuk menyederhanakan proses pelabelan dan pendistribusian data dalam hal pelatihan model YOLOv8, set data yang berisi 26 kelas akan diunggah ke situs web roboflow.com. Proses pelabelan akan melibatkan anotasi *dataset* gambar menggunakan anotasi telapak tangan. Hal ini akan memastikan bahwa model YOLOv8 hanya mendeteksi area tertentu di dalam kotak berlabel. Di Roboflow, kotak anotasi disebut sebagai kotak pembatas. Setelah semua gambar dalam *dataset* diberi label, *dataset* akan disimpan ke direktori penyimpanan Roboflow untuk didistribusikan. Hal ini akan memungkinkan pembagian data untuk pelatihan model YOLOv8, termasuk fase pelatihan, validasi, dan pengujian. Gambar yang telah diberi label dan dikategorikan untuk setiap fase akan disajikan seperti pada gambar di bawah ini.



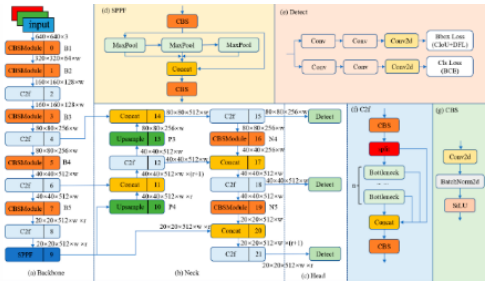
GAMBAR 2.
TAMPILAN DATASET YANG SUDAH DI-*LABELLING* PADA ROBOFLOW

Penggunaan Roboflow akan mengklasifikasikan dataset yang terdiri dari 26 kelas ke dalam data pelatihan, validasi, dan pengujian secara efisien. Seperti terlihat pada gambar di atas, gambar *dataset* telah berhasil dikategorikan. Hasilnya, *dataset* tersebut kini telah siap untuk melatih model YOLOv8.

C. Pelatihan Model

Versi terbaru dari seri YOLO, YOLOv8, memperkenalkan peningkatan substansial dalam desain arsitektur dan efisiensi operasionalnya yang sangat berdampak pada proses pelatihan model. Algoritma ini, yang berasal dari algoritma *Convolutional Neural Network* (CNN), dirancang untuk memproses data jenis gambar, menunjukkan keunggulan dalam klasifikasi gambar, deteksi objek, dan aplikasi tingkat lanjut seperti *Augmented Reality* (AR) dan *virtual reality* (VR). Dalam konteks aplikasi penerjemah bahasa isyarat, khususnya fitur "*Image Translate*" yang menafsirkan gerakan bahasa isyarat SIBI, YOLOv8 digunakan untuk menganalisis data statis yakni alfabet SIBI. Dalam konteks ini, istilah 'statis' menunjukkan bahwa isyarat yang terdiri dari alfabet SIBI tidak tunduk pada pengaruh temporal atau berurutan. Hal ini memungkinkan algoritme YOLO mendeteksi dan menerjemahkan setiap isyarat secara akurat dan efisien.

Algoritma YOLOv8 terdiri dari beberapa jaringan yang terdiri dari tiga komponen berbeda, yakni jaringan tulang punggung, jaringan leher, dan jaringan kepala. Jaringan-jaringan tersebut disusun dalam struktur hierarkis, dimana lapisan awal mendeteksi fitur-fitur sederhana seperti tepi dan garis, kemudian lapisan berikutnya menggabungkan fitur-fitur ini untuk mengenali pola yang lebih kompleks seperti bentuk dan objek. Berikut merupakan gambar ilustrasi struktur arsitektur algoritma pendeteksian objek YOLOv8.



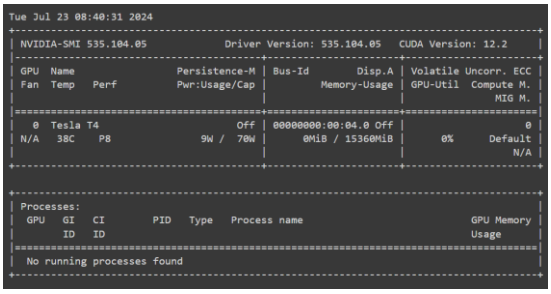
GAMBAR 3.
ARSITEKTUR YOLOV8 MODEL *OBJECT DETECTION*

Model YOLOv8 terdiri dari tiga komponen utama, yakni *backbone*, *neck*, dan *head*. Komponen-komponen ini membentuk struktur inti dari desainnya untuk mendeteksi objek. *Backbone* menggunakan lapisan konvolusi untuk

mengekstrak fitur visual dengan kompleksitas yang berbeda, dari objek dasar hingga objek yang lebih canggih. Ini juga mencakup normalisasi *batch* dan fungsi aktivasi SiLU untuk meningkatkan efisiensi dan stabilitas pelatihan. Lapisan *neck* mengintegrasikan fitur-fitur yang diekstraksi oleh *Backbone* menggunakan teknik fusi multi-skala, seperti jaringan piramida fitur (FPN) dan jaringan agregasi jalur (PAN), untuk meningkatkan resolusi dan meningkatkan representasi fitur untuk mendeteksi objek dengan ukuran dan posisi yang berbeda di dalam gambar [8]. Terakhir, lapisan *Head* memprediksi lokasi dan klasifikasi objek menggunakan lapisan konvolusi pada skala resolusi yang berbeda. Model ini menghasilkan prediksi yang akurat untuk mendeteksi objek dan menghilangkan prediksi yang tidak relevan melalui fungsi aktivasi *sigmoid* dan proses penekanan non-maksimum (NMS)[8].

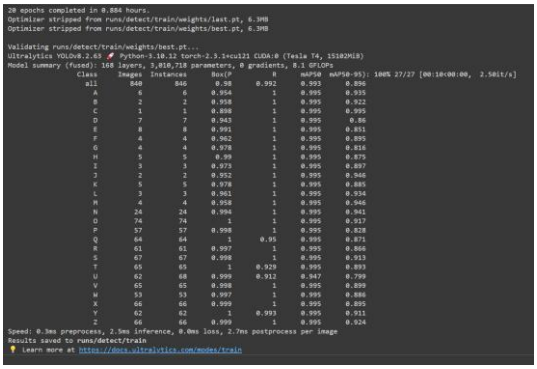
D. Proses Pelatihan Model

Setelah berhasil membuat model, proses pelatihan untuk model pendeteksian objek YOLOv8 dapat dimulai. Langkah penting ini melibatkan pelatihan model untuk mencapai akurasi tertinggi dalam menginterpretasikan alfabet gambar SIBI. Pada bagian berikut ini, dapat dilihat kode sumber yang digunakan untuk pelatihan model dan proses pelatihan algoritma pendeteksian objek YOLOv8. Sebelum memulai pelatihan model YOLOv8, sangat penting untuk memastikan bahwa *runtime* dan akselerator perangkat keras yang digunakan memiliki akses ke GPU. Untuk menggunakan GPU berkinerja tinggi dalam pelatihan model YOLOv8, perintah "*nvidia-smi*" dapat digunakan untuk memverifikasi ketersediaan GPU [9]. Sistem penerjemah bahasa isyarat disini menggunakan GPU Nvidia Tesla 4 bersama dengan CPU Intel Xeon 2.20 GHz. Gambar di bawah ini menunjukkan koneksi yang berhasil ke GPU sebelum proses pelatihan model dimulai.



GAMBAR 4.
BERHASIL TERHUBUNG DENGAN GPU

Dalam proses pelatihan model YOLOv8, *library* Ultralytics perlu di-*install* sebagai prasyarat. Selanjutnya, *library* ini mengembangkan algoritme untuk deteksi objek, termasuk algoritme YOLO.

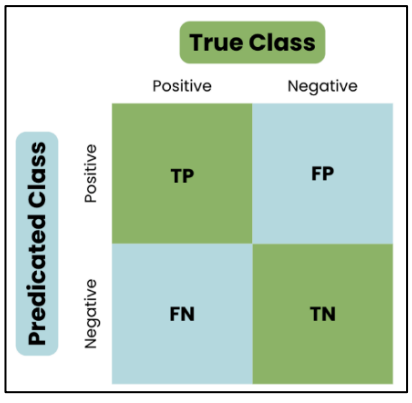


GAMBAR 5.
HASIL PELATIHAN YOLOV8 MODEL *OBJECT DETECTION*

Total durasi proses pelatihan untuk 20 *epoch* adalah 0,884 jam, yang setara dengan 53 menit dan 4 detik. Lamanya proses pelatihan disebabkan karena model dilatih dari kelas A hingga Z, dengan 80% *dataset* digunakan untuk pelatihan dan 20% sisanya dialokasikan untuk validasi dan pengujian. Seperti yang diilustrasikan pada gambar di atas, model YOLOv8 menunjukkan pelatihan yang sukses untuk semua kelas alfabet SIBI, dengan nilai rata-rata *mAP*50 sebesar 0,994 dan nilai *mAP*50-95 sebesar 0,90.

E. Evaluasi Model

Proses pengujian akurasi algoritma YOLOv8 menggunakan matriks evaluasi berdasarkan analisis *confusion matrix*. Matriks ini digunakan untuk menilai keampuhan model klasifikasi dalam *machine learning* dengan membandingkan hasil yang diproyeksikan dari model dengan nilai data aktual [10]. *Confusion matrix* diklasifikasikan ke dalam empat komponen, yakni *true positive* (TP), *true negative* (TN), *false positive* (FP), dan f*alse negative* (FN). Setiap komponen matriks ini memiliki fungsi yang berbeda. TP didefinisikan sebagai jumlah sampel positif yang diklasifikasikan dengan benar. TN adalah jumlah sampel negatif yang diklasifikasikan dengan benar. FP adalah jumlah sampel negatif yang salah diklasifikasikan tetapi dianggap positif, sedangkan FN adalah jumlah sampel positif yang salah diklasifikasikan sebagai negatif.



GAMBAR 6.
ARSITEKTUR *CONFUSION MATRIX*

Selanjutnya, matriks evaluasi mampu menghitung nilai untuk *recall*, presisi, F1-*score,* dan akurasi. Nilai-nilai ini digunakan untuk menilai kinerja model sehingga

memudahkan evaluasi keefektifan. Nilai *recall* didefinisikan sebagai rasio prediksi positif yang benar terhadap prediksi positif yang sebenarnya. Persamaan berikut ini digunakan untuk menghitung nilai *recall:*

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}, \quad (1)$$

Nilai presisi didefinisikan sebagai rasio prediksi positif yang benar terhadap jumlah total prediksi positif. Untuk memastikan nilai presisi, persamaan berikut dapat digunakan:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}, \quad (2)$$

Nilai F1-*Score* diperoleh dengan menghitung rata-rata antara *recall* dan *precision*. Nilai F1-*Score* berada pada rentang 0.00 hingga 1.00, dengan nilai F1-*Score* terbaik adalah 1.00 dan nilai terburuk adalah 0.00. Persamaan untuk menghitung nilai F1-*Score* adalah sebagai berikut:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \quad (3)$$

Nilai akurasi digunakan untuk memastikan jumlah total klasifikasi yang akurat dibagi dengan jumlah total kasus. Persamaan berikut ini digunakan untuk menghitung nilai akurasi model YOLOv8:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Untuk menentukan akurasi model YOLO, evaluasi matriks dari *confusion matrix* tidak menjadi satu-satunya acuan yang digunakan. Nilai *mAP*50 dan *mAP* (50-95) juga dapat digunakan sebagai parameter untuk mengevaluasi kualitas prediksi yang dibuat oleh model YOLO. Parameter *mAP* 50 adalah matriks yang mengukur ketepatan deteksi objek pada ambang batas *Intersection Over Union* (IOU) sebesar 0,50. Matriks *mAP*50-95 memungkinkan evaluasi deteksi objek di berbagai ambang batas IOU, mulai dari 0,50 hingga 0,95, dengan kenaikan 0,05 [11].

IV. HASIL DAN PEMBAHASAN

A. Pengujian algoritma YOLOv8 pada deteksi gambar bahasa isyarat

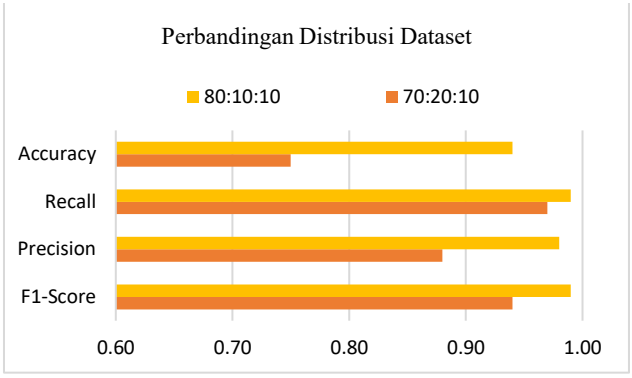
Tujuan dari pengujian akurasi algoritma YOLOv8 adalah untuk mengidentifikasi model YOLOv8 yang optimal, yang kemudian akan digunakan sebagai model *machine learning* untuk fitur penerjemahan gambar pada situs web sistem penerjemah bahasa isyarat bahasa Indonesia. Berikut merupakan hasil pengujian akurasi algoritma YOLOv8 yang dilakukan.

TABEL 3.
HASIL PENGUJIAN DISTRIBUSI DATASET

Distribusi Data	Akurasi	*Recall*	*Precision*	F1-*Score*
80:10:10	0.94	0.99	0.98	0.99
70:20:10	0.75	0.97	0.88	0.93

1. Perbandingan Distribusi *Dataset* pada *train-valid-test*

Objektif dari perbandingan ini adalah untuk memastikan rasio *dataset* dengan akurasi yang lebih unggul dan menghindari potensi masalah *overfitting* atau *underfitting* pada model YOLOv8. Bagian berikut ini menyajikan hasil pengujian untuk perbandingan distribusi *dataset* dalam konteks model pendeteksian objek YOLOv8.

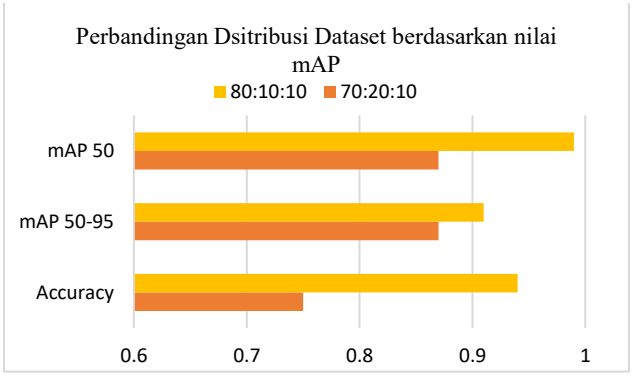


GAMBAR 7.
VISUALISASI PERBANDINGAN DISTRIBUSI *DATASET*

Hasil dari pengujian yang dilakukan menunjukkan bahwa skenario 1 lebih efektif daripada skenario 2. Seperti yang ditunjukkan pada gambar 7, pengujian skenario 1 menunjukkan tingkat akurasi sebesar 0.94, sedangkan skenario 2 menunjukkan nilai akurasi sebesar 0.74. Untuk analisis perbandingan parameter lainnya, pengujian 1 menunjukkan nilai *recall* sebesar 0.99, presisi sebesar 0.98, dan F1-*score* sebesar 0.99. Sebaliknya, pengujian skenario 2 menghasilkan nilai *recall* sebesar 0.97, presisi sebesar 0.88, dan F1-*score* sebesar 0.93. Hasil pengujian tersebut menunjukkan bahwa skenario 1 lebih efektif daripada skenario 2. Selanjutnya, perbandingan jenis distribusi *dataset* disajikan berdasarkan nilai *mAP*50 dan *mAP*50-95.

TABEL 4.
PERBANDINGAN DISTRIBUSI DATASET PADA NILAI *MAP*

Distribusi Data	Akurasi	*mAP* 50	*mAP* 50-95
80:10:10	0.94	0.99	0.91
70:20:10	0.75	0.87	0.87



GAMBAR 8.
VISUALISASI PERBANDINGAN DISTRIBUSI *DATASET* BERDASARKAN NILAI *mAP*

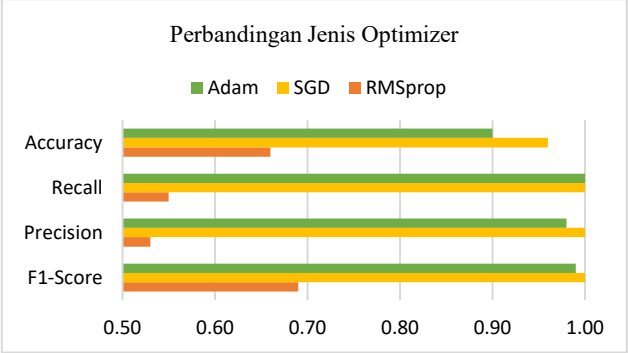
Seperti yang dapat diamati dari data yang disajikan pada tabel 4, nilai skenario pengujian 1 adalah 0.99 *mAP*50, 0.91 *mAP*50-95 dan 0.94 akurasi. Hal ini menunjukkan hasil pengujian skenario 1 lebih tinggi daripada skenario 2. Oleh karena itu, skenario pengujian 1 akan digunakan untuk membandingkan distribusi dataset, di mana 80% data dialokasikan untuk pelatihan, 10% untuk validasi, dan 10% sisanya untuk pengujian.

2. Perbandingan Jenis *Optimizer* pada nodal YOLOv8

Tujuan dari perbandingan *optimizer* adalah untuk menilai keandalan relatif dari berbagai *optimizer* yang berbeda dalam memproses *dataset* gambar bahasa isyarat yang digunakan untuk pelatihan model YOLOv8. Bagian berikut ini menyajikan hasil perbandingan akurasi jenis *optimizer* yang digunakan dalam model YOLOv8.

TABEL 5.
PERBANDINGAN JENIS *OPTIMIZER*

Optimizer	*Accuracy*	*Recall*	*Precision*	F1-*Score*
Adam	0.90	1.00	0.98	0.99
SGD	0.96	1.00	1.00	1.00
RMSprop	0.66	0.55	0.60	0.69



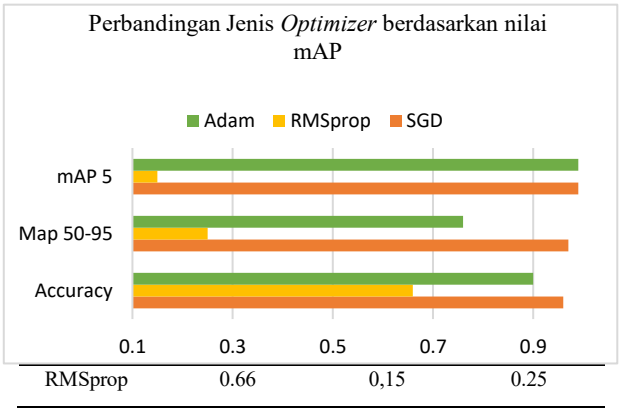
GAMBAR 9.
VISUALISASI PERBANDINGAN JENIS *OPTIMIZER*

Pada skenario pertama, pengujian dengan menggunakan tipe *optimizer* 'Adam' memberikan hasil yang memuaskan, dengan nilai akurasi 0.90, *recall* 1.00, presisi 0.98, dan F1-*score* 0.99. Selain itu, gambar *confusion matrix* menunjukkan bahwa model YOLOv8 pada tipe *optimizer* 'Adam' dapat memprediksi semua kelas A-Z secara akurat. Namun, ada beberapa kasus salah prediksi, khususnya untuk kelas 'D', 'I', 'M' dan 'N'. Sebaliknya, pengujian skenario 2 yang menggunakan jenis *optimizer* 'SGD' menghasilkan hasil yang lebih baik, dengan nilai akurasi 0.96, *recall* 1.00, presisi 1.00, dan F1-*score* 1.00. Terlihat bahwa nilai akurasi pada pengujian skenario 2 lebih besar dibandingkan dengan skenario 1. Selain itu, pengujian skenario 3 dengan menggunakan jenis *optimizer* 'RMSprop' memberikan hasil yang kurang optimal. Hal ini dibuktikan dengan nilai akurasi terendah di antara kedua *optimizer* lainnya, yaitu 0,66. Hal ini dibandingkan dengan nilai parameter lainnya, termasuk *recall* 0.55, *precision* 0.60, dan F1-*score* 0.69. Hasil pengujian *optimizer* menunjukkan bahwa dua tipe yang paling efektif adalah 'Adam' dan 'SGD'.

Selanjutnya, pengujian *optimizer* juga dapat dilakukan lewa perbandingan nilai *mAP*50 dan *mAP*50-95, sehingga penilaian kapasitas model untuk memprediksi kelas alfabet SIBI lebih mudah dilakukan. Berikut ini adalah hasil perbandingan jenis *optimizer* oleh nilai *mAP*50 dan *mAP*50-95, setelah proses pelatihan model YOLOv8 selesai.

TABEL 6.
PERBANDINGAN JENIS *OPTIMIZER* BERDASARKAN NILAI *mAP*

Optimizer	Akurasi	*mAP* 50	*mAP* 50-95
Adam	0.90	0.99	0.76
SGD	0.96	0.99	0.97



GAMBAR 10.
VISUALISASI PERBANDINGAN JENIS OPTIMIZER

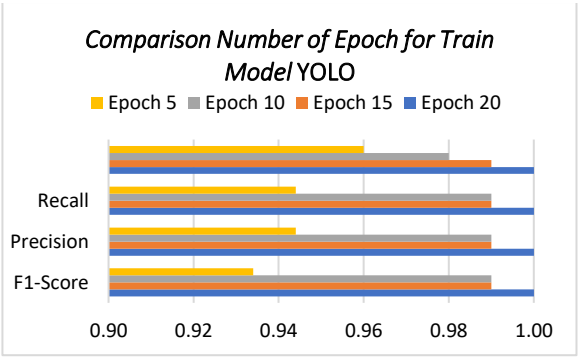
Tabel di atas menunjukkan bahwa model YOLOv8 dengan *optimizer* 'SGD' menunjukkan peningkatan akurasi dan nilai *mAP* dibandingkan dengan *optimizer* 'Adam'. Sebaliknya, *optimizer* 'RMSprop' tidak cocok untuk prediksi YOLOv8 karena akurasi dan nilai *mAP*nya berkurang secara signifikan dibandingkan dengan 'Adam' dan 'SGD'. Perbandingan lebih lanjut dari *optimizer* menunjukkan bahwa *optimizer* 'SGD' menunjukkan akurasi dan *mAP* yang optimal.

3. Perbandingan Jumlah *Epoch Train model* YOLOv8

Jumlah *epoch* yang digunakan dalam perbandingan berfungsi untuk mencegah model YOLOv8 menunjukkan kecenderungan *overfitting* atau *underfitting*. Untuk memastikan potensi akurasi YOLO, pertimbangan nilai *mAP*50 dan *mAP*50-95 perlu dilakukan. Bagian berikut ini menyajikan hasil dari skenario pengujian *epoch* dalam mengidentifikasi nilai akurasi optimal untuk model YOLOv8.

TABEL 7.
PERBANDINGAN JUMLAH *EPOCH* PELATIHAN MODEL YOLOv8

Epoch	Akurasi	*Recall*	*Precision*	F1-*Score*	Durasi Pelatihan
5	0.96	0.94	0.94	0.93	0.192 jam
10	0.98	0.99	0.99	0.99	0.367 jam
15	0.99	0.99	0.99	0.99	0.561 jam
20	1.00	1.00	1.00	1.00	0.759 jam



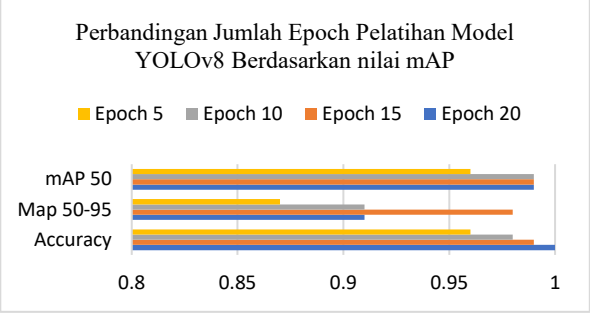
GAMBAR 11.
VISUALISASI PERBANDINGAN JUMLAH *EPOCH* PELATIHAN MODEL YOLOv8

Pada pengujian skenario 1, penggunaan 5 *epoch* terbukti memberikan hasil yang baik, dengan nilai akurasi sebesar 0.96, *recall* sebesar 0.94, *precision* sebesar 0.94, dan F1-

score sebesar 0.93. Proses pelatihan yang dilakukan dengan 5 *epoch* membutuhkan waktu 0.192 jam (setara dengan 11 menit 52 detik). Sebaliknya, pengujian skenario 2 dengan 10 *epoch* memberikan hasil yang lebih baik dengan perolehan nilai akurasi 0.98, serta matriks *recall* (0.99), *precision* (0.99), dan F1-*score* (0.99) yang optimal. Durasi pengujian skenario 2 adalah 0.367 jam (22 menit dan 2 detik), sedikit lebih lama dibandingkan dengan skenario pengujian 1. Skenario pengujian ketiga dengan 15 *epoch* menunjukkan peningkatan nilai akurasi, *recall,* presisi, dan F1-*score,* melebihi nilai yang diamati pada pengujian sebelumnya (0,99). Pada pengujian keempat dengan 20 *epoch*, muncul hasil yang tak terduga. Nilai akurasi, *recall*, presisi, dan F1-*score* mencapai nilai 1.00, yang menunjukkan bahwa model menunjukkan *overfitting* pada *epoch* 20.

TABEL 8.
PERBANDINGAN JUMLAH *EPOCH* PELATIHAN MODEL YOLOv8 BERDASARKAN NILAI mAP

Epoch	Accuracy	*mAP* 50	*mAP* 50-95
5	0.96	0.96	0.87
10	0.98	0.99	0.91
15	0.99	0.99	0.98
20	1.00	0.99	0.96



GAMBAR 12.
VISUALISASI PERBANDINGAN JUMLAH *EPOCH* PADA PELATIHAN MODEL BERDASARKAN NILAI mAP

Dari data yang disajikan pada tabel 8, dapat dilihat bahwa skenario 3 menunjukkan kinerja yang paling optimal dibandingkan dengan skenario lainnya. Hal ini dibuktikan dengan nilai akurasi 0.99, *mAP*50 sebesar 0.99, dan *mAP*50-95 sebesar 0.98. Untuk skenario 4, model telah menunjukkan tanda-tanda *overfitting,* yang dibuktikan dengan nilai akurasi 1,00. Hal ini kontras dengan nilai *mAP*50 yang tetap konsisten dengan yang diamati pada pengujian skenario 3 (0,99). Namun, nilai *mAP*50-95 berbeda karena menunjukkan penurunan sebesar 2% dibandingkan dengan skenario 3 (0,96).

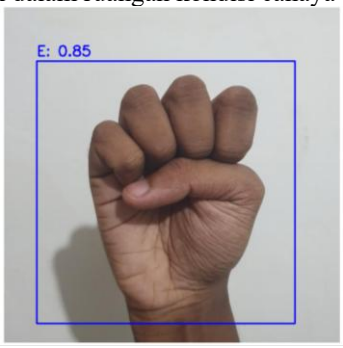
B. Pengujian eksternal algoritma YOLOv8 pada deteksi gambar bahasa isyarat

Tujuan dari pengujian eksternal ini adalah untuk memastikan keampuhan model YOLOv8 dalam konteks lingkungan yang beragam. Penilaian ini dilakukan untuk menjamin bahwa model YOLOv6 merupakan pilihan yang optimal dan siap untuk diimplementasikan pada situs web penerjemah bahasa isyarat. Evaluasi ini akan mencakup dampak cahaya, jarak, dan kualitas gambar terhadap hasil terjemahan.

1. Pengujian pada intensitas cahaya

Tujuan percobaan ini adalah untuk memastikan keampuhan model YOLOv8 dalam memprediksi kelas alfabet SIBI pada gambar yang menunjukkan pencahayaan, saturasi, dan kontras warna yang optimal. Oleh karena itu, perangkat yang dikenal sebagai *luxmeter* diperlukan untuk pengukuran intensitas cahaya, baik di dalam maupun di luar ruangan. *Luxmeter* dapat diperoleh dengan mengunduh aplikasi yang relevan dan tersedia di Google Play Store untuk perangkat Android serta App Store untuk perangkat iOS. Bagian berikut ini menyajikan temuan-temuan dari investigasi mengenai dampak kondisi pencahayaan terhadap kinerja model YOLOv8 dalam sistem penerjemah bahasa isyarat.

a. Pengujian dalam ruangan kondisi cahaya terang



GAMBAR 13.
PENGUJIAN DALAM RUANGAN KONDISI CAHAYA TERANG

Pengujian dilakukan pada ruangan dengan intensitas cahaya rata-rata 882 LUX, maksimum 1190 LUX, dan minimum 240 LUX. Dari pengujian yang dilakukan pada kelas "E", dapat diamati pada gambar 13 bahwa model YOLOv8 secara akurat memprediksi kelas "E" dengan tepat di ruangan dengan kondisi cahaya yang terang. Model YOLO menunjukkan tingkat akurasi yang tinggi, dengan nilai kepercayaan sebesar 0,85.

b. Pengujian dalam ruangan kondisi cahaya redup

GAMBAR 14.
PENGUJIAN DALAM RUANGAN KONDISI CAHAYA REDUP

Pengujian dilakukan di ruangan dengan intensitas cahaya rata-rata 333 LUX, maksimum 830 LUX, dan minimum 0 LUX. Pengujian di atas menunjukkan bahwa model YOLOv8 mampu memprediksi kelas "E" secara akurat dengan nilai kepercayaan sebesar 0,81. Terjadi penurunan 4% dari hasil yang diperoleh pada ruangan dengan kondisi cahaya yang terang.

c. Pengujian luar ruangan kondisi cahaya terang

GAMBAR 15.
PENGUJIAN LUAR RUANGAN DALAM KONDISI CAHAYA TERANG

Gambar di atas memberikan bukti bahwa model YOLOv8 mampu memprediksi alfabet SIBI secara akurat dalam kondisi luar ruangan dengan cahaya terang. Intensitas cahaya rata-rata adalah 2131 LUX, maksimum 5570 LUX dan minimum 120 LUX. Hasilnya, model YOLOv8 berhasil mendeteksi kelas alfabet SIBI "E" dengan nilai kepercayaan 0,85. Namun, nilai kepercayaan tersebut tidak terlihat pada gambar karena ukuran "kotak putih" yang berbeda dibandingkan dengan kotak pembatas model YOLOv8.

d. Pengujian luar ruangan kondisi cahaya redup

GAMBAR 16.
PENGUJIAN LUAR RUANGAN DALAM KONDISI CAHAYA REDUP

Pada pengujian di luar ruangan dengan kondisi cahaya rendah (intensitas cahaya rata-rata 25 LUX, maksimum 46 LUX, dan minimum 0 LUX), model YOLOv8 menunjukkan kesulitan dalam memprediksi kelas alfabet SIBI sehingga hasil yang didapatkan tidak memuaskan. Pengujian di atas menggunakan gambar dari kelas "E", namun, ketika model YOLOv8 mencoba memprediksi, model tersebut tidak dapat secara pasti memastikan bahwa gambar tersebut termasuk dalam kelas "E". Menurut model tersebut, gambar dikategorikan ke dalam dua kelas yakni "E" dan "M". Hal ini menyebabkan kebingungan dan mengurangi akurasi model YOLOv8.

2. Pengujian pada pengaruh jarak

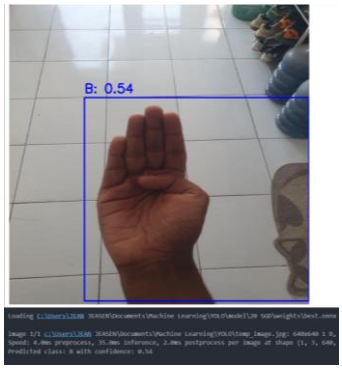
Tujuan dari pengujian jarak adalah untuk memastikan kapasitas model YOLOv8 membuat prediksi dalam jarak dekat atau jarak tertentu. Bagian berikut ini menyajikan hasil dari beberapa pengujian jarak yang dilakukan untuk menilai keampuhan model YOLOv8.

a. Pengujian jarak dekat (0-3 cm)

GAMBAR 17.
PENGUJIAN JARAK DEKAT

Model YOLOv8 diuji pada jarak 0-3 cm dari kamera, seperti pada gambar 17. Pengujian ini juga dilakukan di luar ruangan dengan intensitas cahaya yang cukup. Model YOLOv8 berhasil mendeteksi gambar di ataa sebagai kelas alfabet SIBI "B" dengan nilai kepercayaan sebesar 0,93.

b. Pengujian jarak menengah (4-10 cm)



GAMBAR 18.
PENGUJIAN JARAK MENENGAH

Pengujian jarak dilanjutkan dengan menambahkan jarak 4-10 cm dari kamera. Seperti yang terlihat pada gambar 18, model YOLOv8 berhasil memprediksi kelas "B", tetapi memiliki nilai kepercayaan yang sangat rendah yaitu 0,54. Hal ini sangat kontras dengan hasil yang diperoleh pada jarak 0-3 cm dengan nilai kepercayaan sebesar 0,93 sehingga terjadi penurunan nilai yang cukup besar yaitu 39%.

c. Pengujian jarak jauh (11-20cm)

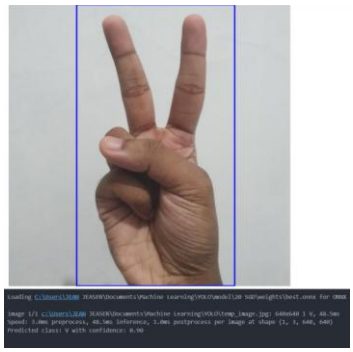
GAMBAR 19.
PENGUJIAN JARAK JAUH

Ketika diuji pada jarak antara 11-20 sentimeter, model YOLOv8 tidak dapat membuat prediksi apa pun. Hal ini terjadi meskipun kotak pembatas tidak berhasil dimuat oleh model YOLOv8, dan nilai kepercayaan tidak ditampilkan.

3. Pengujian pengaruh kemiringan gambar

Penelitian ini menggunakan pengujian kemiringan gambar untuk memastikan sejauh mana model YOLOv8 mampu mendeteksi alfabet kelas SIBI dalam kondisi di mana gambar mengalami kemiringan beberapa derajat. Bagian berikut ini menyajikan analisis komparatif pengujian kemiringan gambar yang dilakukan untuk mengevaluasi model pendeteksian objek YOLOv8.

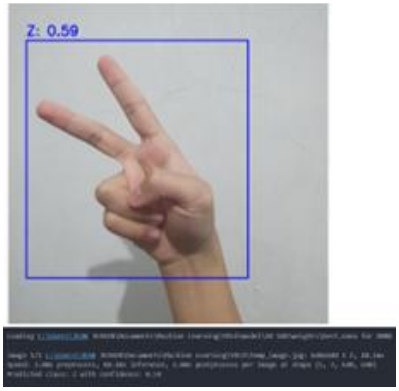
a. Pengujian kemiringan pada 0°



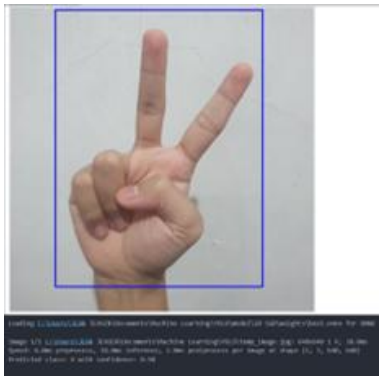
GAMBAR 20.
PENGUJIAN PADA KEMIRINGAN 0°

Pada uji kemiringan gambar 0°, model YOLOv8 menunjukkan klasifikasi yang akurat untuk kelas “V”, dengan nilai kepercayaan 0,90.

b. Pengujian kemiringan pada 30°



GAMBAR 21.
PENGUJIAN KEMIRINGAN PADA 30° KE ARAH KIRI

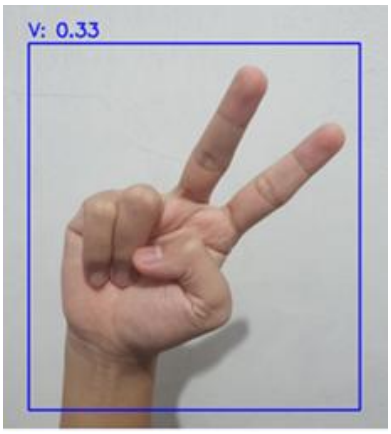


GAMBAR 22
PENGUJIAN KEMIRINGAN PADA 30° KE ARAH KIRI

Pada uji kemiringan gambar 30 derajat, uji kemiringan gambar dua kali lipat dilakukan ke kiri dan ke kanan. Hasilnya menunjukkan bahwa model YOLOv8 mengidentifikasi kelas “V” sebagai kelas “Z”. Hal ini mengindikasikan adanya potensi masalah dengan persamaan pergerakan, yang dapat menyebabkan kebingungan dalam mengklasifikasikan gambar. Setelah menguji kemiringan gambar pada 30° ke kanan, model YOLOv8 berhasil mengidentifikasi gambar tersebut sebagai kelas “V” dengan nilai kepercayaan 0,90.

c. Pengujian kemiringan pada 60°

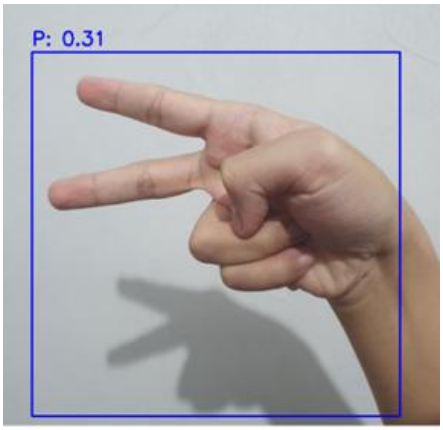
GAMBAR 23.
PENGUJIAN KEMIRINGAN PADA 60° KE ARAH KIRI



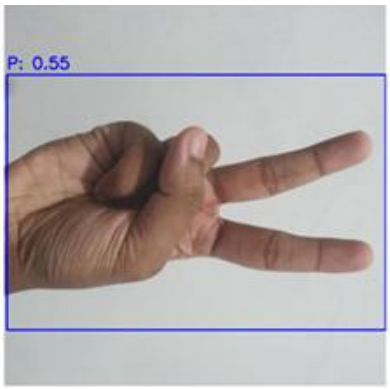
GAMBAR 24.
PENGUJIAN KEMIRINGAN PADA 60° KE ARAH KANAN

Pada pengujian kemiringan gambar 60° ke kiri, model YOLO terus mengidentifikasi kelas “V” sebagai “Z”. Hal ini menunjukkan konsistensi dengan pengujian sebelumnya yakni pengujian dengan kemiringan 30° ke kiri. Selama pengujian ke kanan, model YOLOv8 mampu membuat prediksi yang akurat, meskipun nilai kepercayaannya relatif rendah, yaitu 0,33. Berbeda dengan pengujian sebelumnya yang menghasilkan nilai kepercayaan mencapai 0,90, terjadi penurunan sekitar 63% pada pengujian kemiringan ke kanan 60°.

d. Pengujian kemiringan pada 90°



GAMBAR 25.
PENGUJIAN PADA KEMIRINGAN 90° KE ARAH KIRI



GAMBAR 26.
PENGUJIAN KEMIRINGAN PADA 90° KE ARAH KANAN

Setelah menguji kemiringan gambar pada 90°, baik ke arah kiri maupun kanan, model YOLO mengidentifikasi gambar sebagai kelas "P". Seharusnya, gambar tersebut dikategorikan sebagai kelas "V". Hal ini menunjukkan bahwa ketika kelas "V" berada pada posisi kemiringan 90°, maka sistem akan keliru mendeteksinya sebagai kelas "P". Selain itu, kesamaan posisi antara kedua kelas tersebut dapat menyebabkan kebingungan model YOLOv8 untuk memprediksi kelas SIBI secara akurat.

I. KESIMPULAN

Dari penelitian yang sudah dilakukan, dapat disimpulkan bahwa algoritma YOLOv8 dapat melalukan deteksi gambar pada bahasa isyarat SIBI. Algoritma YOLOv8 menunjukkan kinerja lebih baik dalam mengidentifikasi dan menerjemahkan gambar yang mewakili huruf dalam alfabet bahasa isyarat dengan tingkat akurasi 99% dan *mAP* 50-95 sebesar 98%. Model YOLOv8 mampu mendeteksi alfabet SIBI, baik di dalam maupun luar ruangan, serta kondisi cahaya terang. Pada kondisi intensitas cahaya rendah, model YOLOv8 masih mampu melakukan prediksi kelas alfabet SIBI, meskipun akurasi nya tidak setinggi hasil pada kondisi cahaya terang. Pada kondisi di luar ruangan dengan cahaya redup, model YOLOv8 tidak mampu melakukan prediksi.

Dalam pengujian jarak, model YOLOv8 menunjukkan kinerja optimal pada jarak 0-3 cm. Di luar jarak tersebut, model mulai menunjukkan penurunan akurasi prediksi. Selain itu, pengujian kemiringan gambar menunjukkan bahwa model YOLOv8 mampu memprediksi kelas alfabet SIBI pada kemiringan 0°-60° dengan benar. Namun, model ini menunjukkan penurunan kinerja pada sudut yang melebihi kisaran tersebut. Hal ini mengindikasikan adanya potensi keterbatasan kemampuan sistem untuk mendeteksi gambar alfabet SIBI secara akurat. Penting untuk diingat bahwa setiap alfabet SIBI menunjukkan pergerakan dan posisi yang berbeda. Satu gerakan atau posisi yang identik dengan gerakan atau posisi lainnya, kemungkinan akan menyebabkan kebingungan dalam kemampuan model YOLOv8 untuk mengidentifikasi alfabet SIBI secara akurat.

REFERENSI

[1] J. K. Hikmalansya dan D. Cahyono. "Aplikasi pembelajaran bahasa isyarat berbasis android." *Jurnal INFORM*, vol. 1, no. 2, pp. 118-124, Jul. 2016, doi: 10.25139/inform.v1i2.849. [*Online*]. *Available:* https://ejournal.unitomo.ac.id/index.php/inform/article/view/849

[2] D. Naren, "Aksinya viral saat debat pilpres 2024, ternyata segini gaji fantastis juru bicara isyarat, tertarik?" *Ayobandung.com,* 9 Jan. 2024. [*Online*]. *Available:* https://www.ayobandung.com/umum/7911457619/aksinya-viral-saat-debat-pilpres-2024-ternyata-segini-gaji-fantastis-juru-bicara-isyarat-tertarik

[3] S. Shania, M. F. Naufal, V. R. Prasetyo, dan M. S. Bin Azmi, "Translator of Indonesian sign language video using convolutional neural network with transfer learning," *Indonesian Journal of Information Systems*, vol. 5, no. 1, pp. 17-27, Agu. 2022, doi: 10.24002/ijis.v5i1.5865. [*Online*]. *Available:* https://api.semanticscholar.org/CorpusID:257295702

[4] N. Nurrahma, R. Yusuf, dan A. S. Prihatmanto. "Indonesian sign language fingerspelling recognition using vision-based machine learning," in *2021 International Conf. on Intelligent Cybernetics Technology & Applications (ICICyTA)*, Des. 1-2, 2021. hlm. 28–33, 2021. [*Online*]. *Available:* https://api.semanticscholar.org/CorpusID:246533726

[5] Direktorat Pendidikan Masyarakat dan Pendidikan Khusus. (30 Des. 2020). *Kamus SIBI Kerjasama antara Kementerian Pendidikan dan Kebudayaan dengan Lembaga Penelitian dan Pengembangan Sistem Isyarat Bahasa Indonesia*. [*Online*]. *Available:* https://pmpk.kemdikbud.go.id/sibi/

[6] M. Alaftekin, I. Pacal, dan K. Cicek. "Real-time sign language recognition based on YOLO algorithm," *Neural Comput Appl*, vol. 36, no. 14, pp. 7609–7624, Feb. 2024, doi: 10.1007/s00521-024-09503-6. [*Online*]. *Available:* https://link.springer.com/article/10.1007/s00521-024-09503-6

[7] Ultralytics, "Ultraltics YOLOv8." Ultralytics YOLOv8 Docs, *Accessed:* 14 Agu. 2024. [*Online*]. *Available*: https://docs.ultralytics.com/models/yolov8/#what-is-yolov8-and-how-does-it-differ-from-previous-yolo-versions

[8] L. Shen, B. Lang, dan Z. Song, "DS-YOLOv8-based object detection method for remote sensing images," *IEEE Access*, vol. 11, pp. 125122–125137, Nov. 2023, doi: 10.1109/ACCESS.2023.3330844. [*Online*]. *Available:* https://ieeexplore.ieee.org/iel7/6287639/6514899/10310143.pdf

[9] NVIDIA. "NVIDIA-SMI," NVIDIA. *Accessed:* [*Online*]. *Available*: www.nvidia.com

[10] N. A. Ahmed, "What is a confusion matrix in machine learning? the model evaluation tool explained." Datacamp, *Accessed*: 14 Agu. 2024. [*Online*]. *Available*: https://www.datacamp.com/tutorial/what-is-a-confusion-matrix-in-machine-learning

[11] J. Solawetz, "What is mean average precision (mAP) in object detection?" roboflow, *Accessed:* 14 Agu. 2024. [*Online*]. *Available*: https://blog.roboflow.com/mean-average-precision/#:~:text=Mean%20Average%20Precision%20(mAP)%20is%20used%20to%20measure%20the%20performance,versions%20of%20the%20same%20model.