

SISTEM PENDETEKSIAN BAHASA ISYARAT ALFABET SIBI MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE DAN ROBOFLOW

Jean Jeasen Timotius
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

jeanjeasen@student.telkomuniversity.ac.id

Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

Marisa W. Paryasto
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

marisaparyasto@telkomuniversity.ac.id

Abstrak — 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 teks dan ucapan. Pada penelitian ini akan dikembangkan 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 dari sumber internet dan diambil secara manual. 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. Optimizer SGD memberikan kinerja terbaik dibandingkan optimizer lain seperti Adam dan RMSprop. Sistem ini diharapkan dapat meningkatkan aksesibilitas dan pembelajaran bahasa isyarat di Indonesia.

Kata kunci— Bahasa isyarat, YOLOv8, SIBI, machine learning, object detection, optimizer, mAP

I. PENDAHULUAN

Masalah utama yang dihadapi penyandang disabilitas khususnya tunarungu adalah kurangnya aksesibilitas terhadap informasi dan komunikasi. Orang yang menggunakan bahasa isyarat sebagai cara utama untuk berkomunikasi dapat lebih mudah diakses jika mereka menerjemahkan bahasa isyarat ke dalam teks atau ucapan. Akibatnya, masalah ini menunjukkan betapa pentingnya mencari cara untuk memberi komunitas ini akses yang lebih baik ke informasi. Bahasa isyarat adalah bahasa yang berkomunikasi dengan menggunakan bahasa tubuh, gerak bibir, dan bentuk tangan, serta gerak tangan, dan wajah untuk mengungkapkan pikirannya [1]. Penyandang tunarungu utamanya berkomunikasi melalui bahasa isyarat. Di berbagai belahan dunia, terdapat berbagai jenis bahasa isyarat yang berbeda, misalnya *American Sign Language (ASL)* di Amerika Serikat dan Sistem Isyarat Bahasa Indonesia (SIBI) di Indonesia. Bahasa isyarat membantu penyandang tunarungu berkomunikasi, tetapi masyarakat umum masih sangat terbatas dalam memahami dan menggunakan bahasa ini.

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 melalui media buku pembelajaran bahasa

isyarat juga bisa didapatkan dengan menyewa jasa seorang penerjemah bahasa isyarat, namun cara ini masih terbilang sangat minim dilakukan karena keterbatasan biaya, dibutuhkan biaya Rp 1.500.000 untuk sekali sesi pembelajaran bahasa isyarat[2]. Oleh karena itu dibutuhkan sebuah solusi untuk mengatasi masalah pembelajaran bahasa isyarat yang awalnya hanya terpaku pada metode konvensional dan membutuhkan biaya yang mahal menjadi sebuah metode pembelajaran yang interaktif, minim biaya dan dapat digunakan baik penderita tunarungu dan masyarakat.

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

Pada penelitian penerjemahan bahasa isyarat ini digunakan *machine learning* untuk mempelajari bahasa isyarat indonesia dan menerjemahkan menjadi bahasa Indonesia. Dengan adanya sebuah sistem yang dapat menerjemahkan bahasa isyarat menjadi bahasa indonesia dapat mempercepat proses pembelajaran bahasa isyarat, pembelajaran yang dilakukan bersama sistem dapat menghasilkan tingkat akurasi penerjemah yang akurat sehingga setiap gerakan 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 yang mengembangkan sebuah sistem komputer yang dapat mengerjakan pekerjaan manusia yaitu *machine learning*[3], [4]. Terutama pada penerjemahan bahasa isyarat dapat dilakukan dengan menggunakan *machine learning* untuk melakukan proses pembelajaran dan penerjemahan bahasa isyarat indonesia yang benar, pada bahasa isyarat indonesia SIBI terdapat 4 kelas yaitu: angka, huruf, kata dan imbuhan[5]. Setiap kelas ini memiliki gerakan dan arti yang berbeda-beda oleh karena itu *machine learning* digunakan untuk menerjemahkan bahasa isyarat menjadi bahasa Indonesia. Pada penelitian ini akan memfokuskan untuk meneliti penerjemahan bahasa isyarat SIBI pada kelas huruf yang terdiri dari 26 kelas yaitu huruf A-Z. terdapat beberapa penelitian yang menggunakan algoritma *machine learning* untuk melakukan penerjemahan bahasa isyarat seperti CNN (Convolutional Neural Networks), LSTM (Long Short-Term

Memory), dan YOLO (You Only Look Once). Penelitian serupa 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 kinerja mereka dalam mengenali isyarat tangan dibandingkan. 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. Pada penelitian ini, Roboflow digunakan untuk memudahkan distribusi data, perbandingan, pelabelan data, dan penyimpanan dataset.

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

A. Pengumpulan Dataset

Algoritme YOLOv8 secara eksklusif menggunakan kumpulan data gambar. Alfabet SIBI harus dikonversi ke dalam format gambar agar 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 adalah sebagai berikut: 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, Z. Kumpulan data terdiri dari gambar-gambar yang diambil secara manual yang bersumber dari Kaggle, dengan informasi terperinci tentang sumber gambar yang disertakan dalam setiap kumpulan data yang disediakan pada bagian selanjutnya.

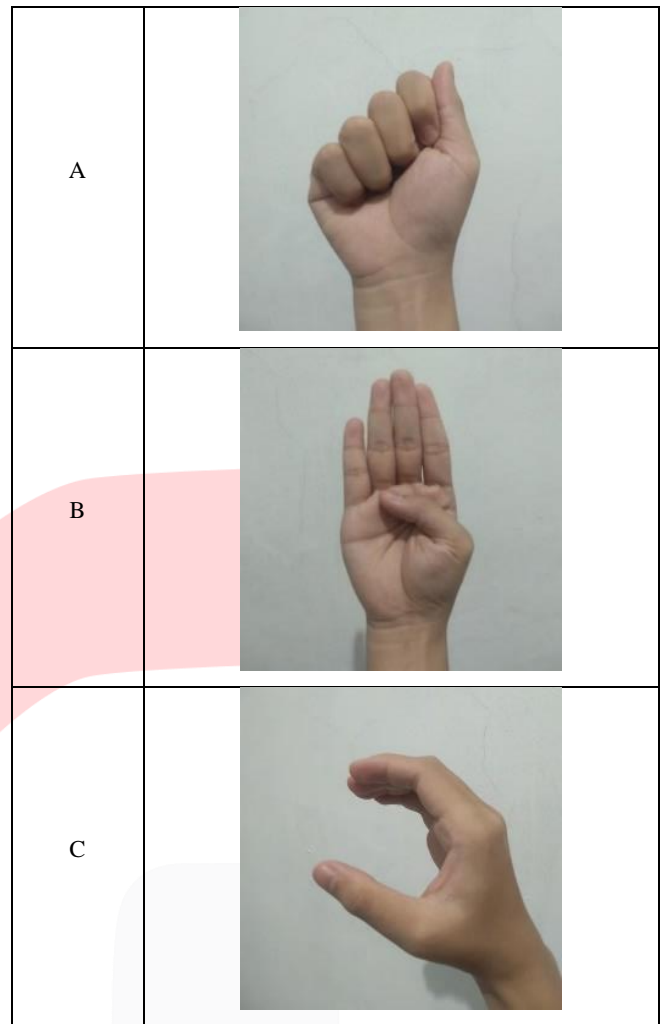
Tabel 1. Detail Informasi Gambar untuk Dataset.

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

Berikut dibawah ini dilampirkan gambar dari beberapa dataset alfabet SIBI:

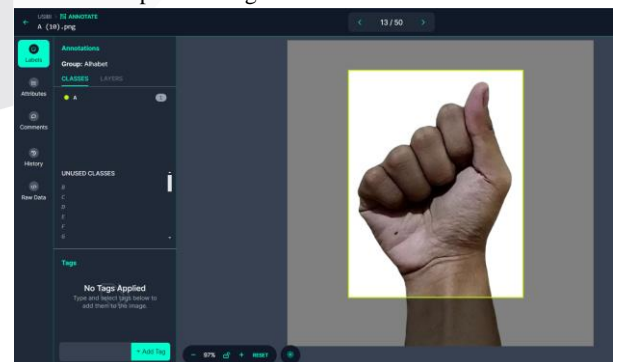
Tabel 2. Contoh Dataset pada Alfabet SIBI.

Huruf	Gambar



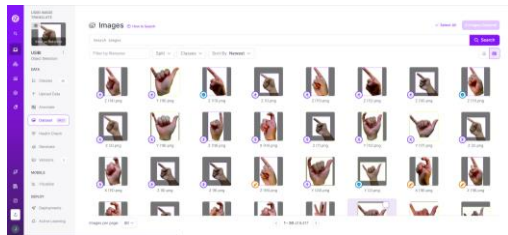
B. Preprocessing Dataset

Pada *prepossessing* dataset, setiap gambar dataset harus berukuran 640x640 piksel. Jika data tidak memenuhi persyaratan ini, data dapat diubah ukurannya secara manual, atau fungsi Python dapat digunakan 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 untuk melabel kumpulan data gambar di Roboflow



Gambar 1. Proses Labelling Dataset pada Roboflow Untuk menyederhanakan proses pelabelan dan pendistribusian data untuk pelatihan model YOLOv8, para penyihir akan mengunggah set data yang berisi 26 kelas 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 di bawah ini.



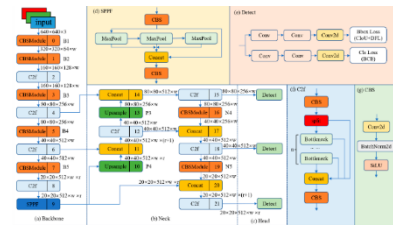
Gambar 2. Tampilan Dataset yang sudah di labelling pada Roboflow

Penggunaan Roboflow secara efisien mengklasifikasikan dataset yang terdiri dari 26 kelas dalam data pelatihan, validasi, dan pengujian. Seperti yang digambarkan 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 operasional, 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 alfabet SIBI, yang merupakan data statis. 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 untuk secara akurat dan efisien mendeteksi dan menerjemahkan setiap isyarat.

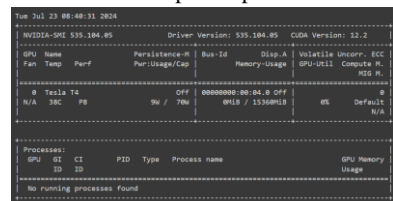
Algoritma YOLOv8 terdiri dari beberapa jaringan, termasuk: Jaringan ini terdiri dari tiga komponen yang berbeda: jaringan tulang punggung, jaringan leher, dan jaringan kepala. Jaringan-jaringan tersebut disusun dalam struktur hierarkis, dengan lapisan awal mendeteksi fitur-fitur sederhana seperti tepi dan garis, dan lapisan berikutnya menggabungkan fitur-fitur ini untuk mengenali pola yang lebih kompleks seperti bentuk dan objek. Gambar berikut mengilustrasikan struktur arsitektur algoritma pendeteksian objek YOLOv8:



Gambar 3. Arsitektur YOLOv8 model *object detection*
 Model YOLOv8 terdiri dari tiga komponen utama: *Backbone*, *Neck*, dan *Head*. Komponen-komponen ini membentuk struktur inti dari desain nya 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. *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 deteksi objek dan menghilangkan prediksi yang tidak relevan melalui fungsi aktivasi sigmoid dan proses penekanan non-maksimum (NMS)[8].

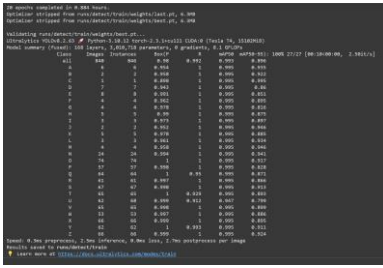
D. Proses Pelatihan Model

Setelah berhasil membuat model, mulailah proses pelatihan untuk model pendeteksian objek YOLOv8. Langkah penting ini melibatkan pelatihan model untuk mencapai akurasi tertinggi dalam menginterpretasikan alfabet gambar SIBI. Pada bagian berikut ini, saya akan memberikan 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 untuk pelatihan model YOLOv8, perintah "nvidia-smi" dapat digunakan untuk memverifikasi ketersediaan GPU[9]. Sistem penerjemah bahasa isyarat kami 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, perlu instal library Ultralytics sebagai prasyarat. Selanjutnya, library ini mengembangkan algoritme untuk deteksi objek, termasuk algoritme YOLO.

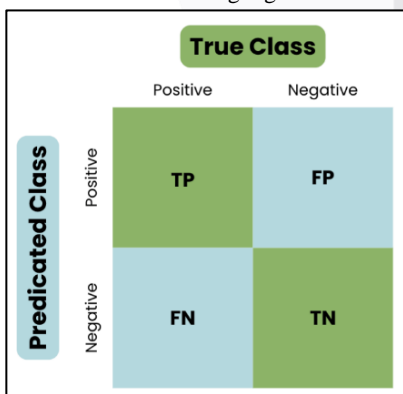


Gambar 5. Hasil Pelatihan YOLOv8 model *object detection*

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 *mAP50* sebesar 0,994 dan nilai *mAP50-95* sebesar 0,90.

E. Evaluasi Model

Proses pengujian akurasi algoritma YOLOv8 menggunakan metrik evaluasi berdasarkan analisis matriks kebingungan. Matriks ini digunakan untuk menilai kemampuan model klasifikasi dalam pembelajaran mesin dengan membandingkan hasil yang diproyeksikan dari model dengan nilai data aktual [10]. *Confusion matrix* diklasifikasikan ke dalam empat komponen: *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Setiap komponen matriks memiliki fungsi yang berbeda. *True positive* (TP) didefinisikan sebagai jumlah sampel positif yang diklasifikasikan dengan benar. Demikian pula, *true negative* (TN) adalah jumlah sampel negatif yang diklasifikasikan dengan benar. *False positive* (FP) adalah jumlah sampel negatif yang salah diklasifikasikan tetapi dianggap positif, sedangkan *false negative* (FN) adalah jumlah sampel positif yang salah diklasifikasikan sebagai negatif. Gambar berikut ini menggambarkan matriks kebingungan:



Gambar 6. Arsitektur *Confusion Matrix*

Selain itu, metrik 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:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}, [10]$$

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

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}, [10]$$

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:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}, [10]$$

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:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} [10]$$

Untuk menentukan akurasi model YOLO, tidak hanya evaluasi metrik dari confusion matrix yang digunakan sebagai acuan. Nilai *mAP50* dan *mAP* (50-95) dapat digunakan sebagai parameter untuk mengevaluasi kualitas prediksi yang dibuat oleh model YOLO. Parameter *mAP* 50 adalah metrik yang mengukur ketepatan deteksi objek pada ambang batas Intersection Over Union (IOU) sebesar 0,50. Metrik *mAP50-95* memungkinkan evaluasi deteksi objek di berbagai ambang batas IOU, dari 0,50 hingga 0,95, dengan kenaikan 0,05 [11].

IV. HASIL DAN PEMBAHASAN

A. Pengujian algoritma YOLOv8 pada deteksi gambar bahasa isyarat

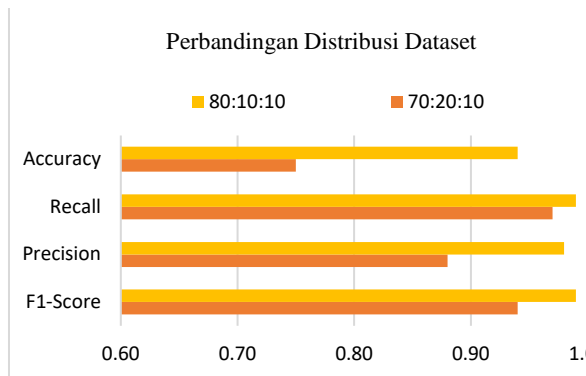
Tujuan dari pengujian akurasi algoritma YOLOv8 adalah untuk mengidentifikasi model YOLOv8 yang optimal, yang kemudian akan digunakan sebagai model pembelajaran mesin untuk fitur penerjemahan gambar pada situs web sistem penerjemah bahasa isyarat bahasa Indonesia. Berikut ini disajikan hasil pengujian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

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

Objektif dari perbandingan ini adalah untuk memastikan rasio dataset mana yang menunjukkan akurasi yang lebih unggul dan untuk 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.

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

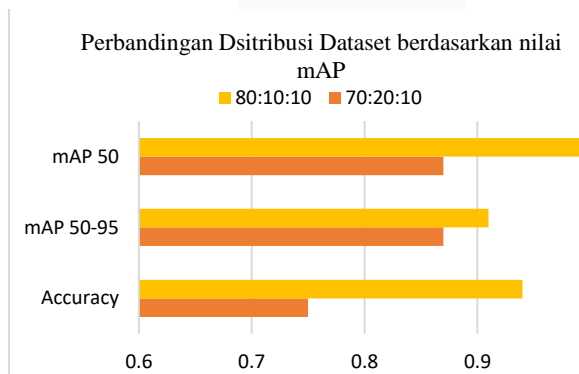


Gambar 7. Visualisasi Perbandingan Distribusi Dataset

Hasil dari pengujian yang dilakukan menunjukkan bahwa skenario 1 lebih efektif daripada skenario 2. Seperti yang ditunjukkan pada tabel perbandingan, pengujian skenario 1 menunjukkan tingkat akurasi sebesar 0.94, sedangkan pengujian skenario 2 menunjukkan nilai akurasi sebesar 0.74. Untuk analisis perbandingan parameter lainnya, pengujian 1 menunjukkan nilai recall sebesar 0.99, nilai presisi sebesar 0.98, dan nilai F1-score sebesar 0.99. Sebaliknya, pengujian skenario 2 menghasilkan nilai recall sebesar 0.97, nilai presisi sebesar 0.88, dan nilai 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 $mAP50$ dan $mAP50-95$.

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, nilai skenario pengujian 1 adalah 0.99 $mAP50$, 0.91 $mAP50-95$ dan 0.94 akurasi, yang lebih tinggi daripada skenario pengujian 2. Oleh karena itu, untuk tujuan membandingkan distribusi dataset, skenario pengujian 1 yang disebutkan di atas akan digunakan, di mana 80% data dialokasikan untuk pelatihan, 10% untuk validasi, dan 10% sisanya untuk pengujian.

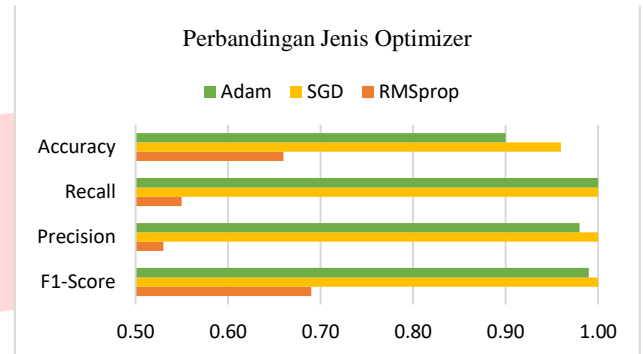
- Perbandingan Jenis *Optimizer* pada nodal YOLOv8

Tujuan dari perbandingan *Optimizer* adalah untuk menilai keandalan relatif dari berbagai *optimizer* yang berbeda untuk memproses set data gambar bahasa isyarat yang

digunakan dalam pelatihan model YOLOv8. Bagian berikut ini menyajikan hasil perbandingan akurasi jenis *optimizer* yang digunakan dalam model YOLOv8,

Tabel 5. Perbandingan Jenis *Optimizer*

<i>Optimizer</i>	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' telah secara akurat memprediksi semua kelas A-Z. Namun, ada beberapa kasus di mana prediksi yang salah telah dibuat, khususnya untuk kelas 'D', 'T', '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, yang dibuktikan dengan nilai akurasi terendah di antara kedua *optimizer*, yaitu 'Adam' dan 'SGD', 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* memungkinkan perbandingan nilai $mAP50$ dan $mAP50-95$, sehingga memudahkan penilaian kapasitas model untuk memprediksi kelas alfabet SIBI. Berikut ini adalah hasil perbandingan jenis *optimizer*, yang ditentukan oleh nilai $mAP50$ dan $mAP50-95$, setelah proses pelatihan model YOLOv8 selesai.

Tabel 6. Perbandingan Jenis *Optimizer* berdasarkan nilai mAP

<i>Optimizer</i>	Akurasi	$mAP 50$	$mAP 50-95$
Adam	0.90	0.99	0.76
SGD	0.96	0.99	0.97

RMSprop	0.66	0.15	0.25
---------	------	------	------

Gambar 10. Visualisasi Perbandingan Jenis Optimizer

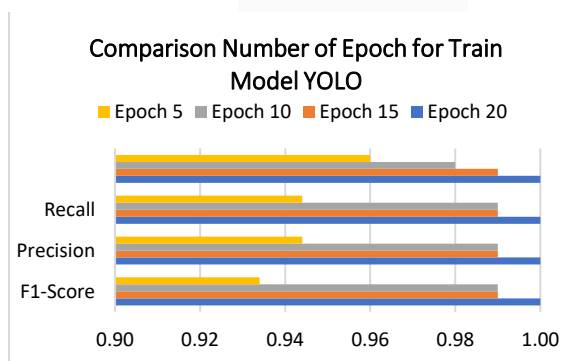
Tabel di atas menunjukkan bahwa model YOLOv8 yang menggunakan optimizer 'SGD' menunjukkan peningkatan akurasi dan nilai *mAP* dibandingkan dengan optimizer 'Adam'. Sebaliknya, optimizer 'RMSprop' tidak cocok untuk prediksi YOLOv8 karena akurasi dan nilai *mAP* yang berkurang secara signifikan dibandingkan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

- 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, kita harus mempertimbangkan nilai *mAP*50 dan *mAP*50-95. Bagian berikut ini menyajikan hasil dari skenario pengujian epoch yang dilakukan untuk 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



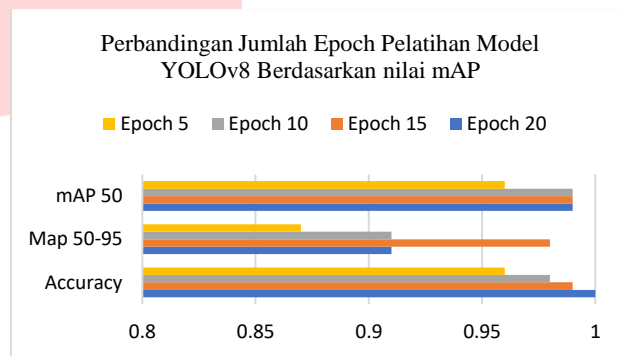
Gambar 11. Visualisasi Perbandingan Jumlah Epoch Pelatihan Model YOLOv8

Pada pengujian skenario 1, penggunaan 5 epoch telah diamati memberikan hasil yang baik, dengan nilai akurasi sebesar 0.96, nilai recall sebesar 0.94, nilai precision sebesar 0.94, dan nilai F1-score sebesar 0.93. Proses pelatihan yang dilakukan dengan lima 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 model pelatihan YOLOv8 yang menunjukkan nilai akurasi sebesar 0.98, serta metrik 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 menggunakan 15 epoch, menunjukkan peningkatan nilai akurasi, recall, presisi, dan F1-score, melebihi nilai yang diamati pada pengujian sebelumnya (0.99). Pada pengujian keempat, yang menggunakan 20 epoch, hasil yang tidak terduga teramati. 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	<i>mAP</i> 50	<i>mAP</i> 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



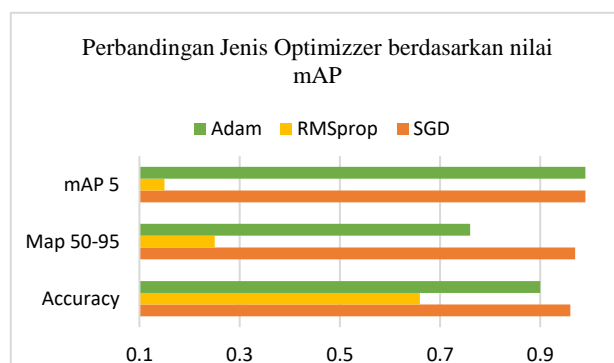
Gambar 12. Visualisasi Perbandingan Jumlah Epoch Pada Pelatihan Model berdasarkan nilai *mAP*

Dari data yang disajikan pada tabel, dapat dilihat bahwa skenario 3 menunjukkan kinerja yang paling optimal dibandingkan dengan skenario lainnya. Hal ini dibuktikan dengan nilai akurasi sebesar 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 sebesar 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, menunjukkan penurunan sebesar 2% dibandingkan dengan skenario 3 (0.96).

B. Pengujian eksternal algoritma YOLOv8 pada deteksi gambar bahasa isyarat

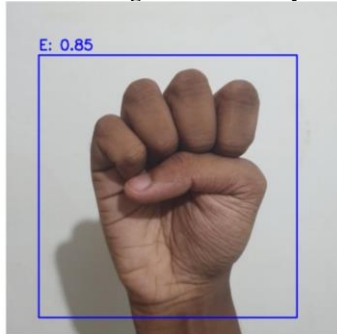
Tujuan dari pengujian eksternal ini adalah untuk memastikan kemampuan model YOLOv8 dalam konteks lingkungan yang beragam. Penilaian ini dilakukan untuk menjamin bahwa model YOLOv6 merupakan pilihan yang optimal dan siap untuk diimplementasikan pada situs web penerjemah bahasa isyarat. Evaluasi ini akan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

1. Pengujian pada intensitas cahaya



Adapun 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, tersedia di Google Play Store untuk perangkat Android dan 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.

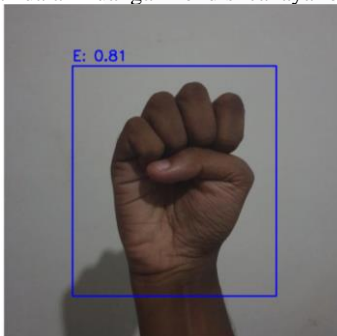
- Pengujian dalam ruangan kondisi cahaya terang



Gambar 13. Pengujian dalam ruangan kondisi cahaya terang

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

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

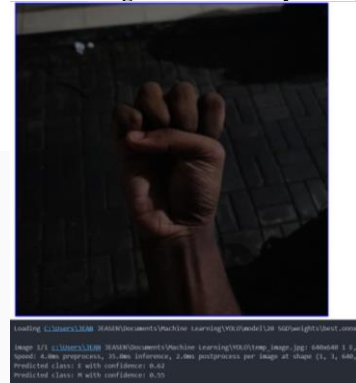
- Pengujian luar ruangan kondisi cahaya terang



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 di luar ruangan dengan cahaya yang terang. Intensitas cahaya rata-rata adalah 2131 LUX, dengan 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.

- 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, minimum 0 LUX), model YOLOv8 menunjukkan kesulitan dalam memprediksi kelas alfabet SIBI, yang mengakibatkan hasil yang tidak memuaskan. Pengujian yang disebutkan 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 sebagai milik dua kelas: "E" dan "M". Hal ini menyebabkan kebingungan dan mengurangi akurasi model YOLOv8.

2. Pengujian pada pengaruh jarak

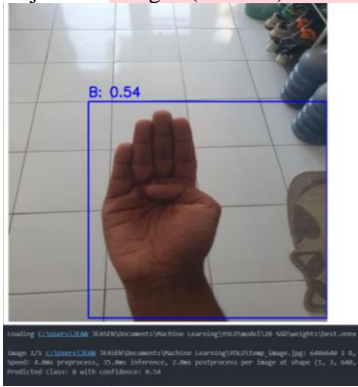
Tujuan dari pengujian jarak adalah untuk memastikan kapasitas model YOLOv8 dalam membuat prediksi pada jarak dekat atau jarak tertentu. Bagian berikut ini menyajikan hasil dari beberapa pengujian jarak yang dilakukan untuk menilai keampuhan model YOLOv8.

- Pengujian jarak dekat (0-3 cm)



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

- Pengujian jarak menengah (4-10 cm)



Gambar 18. Pengujian jarak menengah Pengujian jarak dilanjutkan dengan menambahkan jarak 4 hingga 10 cm dari kamera. Seperti yang dapat dilihat pada gambar di atas, model YOLOv8 berhasil memprediksi kelas "B", tetapi memiliki nilai kepercayaan yang sangat rendah yaitu 0,54. Sangat kontras dengan hasil yang diperoleh pada jarak 0-3 cm, yang menghasilkan nilai kepercayaan sebesar 0,93, terjadi penurunan nilai yang cukup besar yaitu 39%.

- Pengujian jarak jauh (11-20cm)

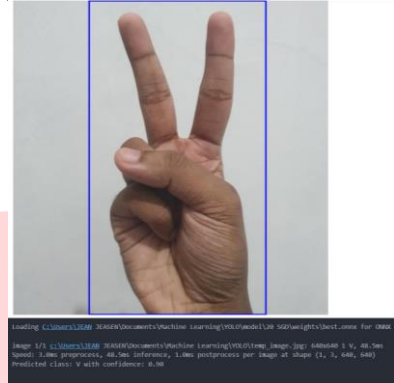


Gambar 19. Pengujian jarak jauh Ketika diuji pada jarak antara 11 dan 20 sentimeter, model YOLOv8 tidak dapat membuat prediksi apa pun. Hal ini terjadi meskipun kotak pembatas tidak berhasil dimuat oleh model YOLOv8, dan nilai kepercayaan tidak ditampilkan.

3. Pengujian pengaruh kemiringan gambar

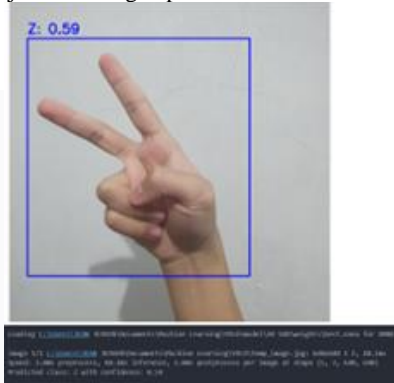
Penelitian ini menggunakan pengujian kemiringan gambar untuk memastikan sejauh mana model YOLOv8 mampu mendeteksi alfabet kelas SIBI dalam kondisi di mana gambar mengalami kemiringan beberapa derajat. Bagian berikut ini menyajikan analisis komparatif pengujian kemiringan citra yang dilakukan untuk mengevaluasi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

- Pengujian kemiringan pada 0°

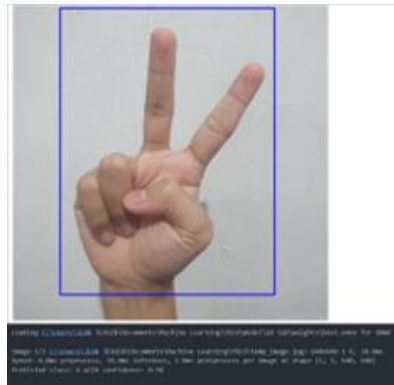


Gambar 20. Pengujian pada kemiringan 0° Pada uji kemiringan gambar 0°, model YOLOv8 menunjukkan klasifikasi yang akurat untuk kelas "V", dengan nilai kepercayaan 0,90.

- Pengujian kemiringan pada 30°



Gambar 21. Pengujian kemiringan pada 30° ke arah kiri

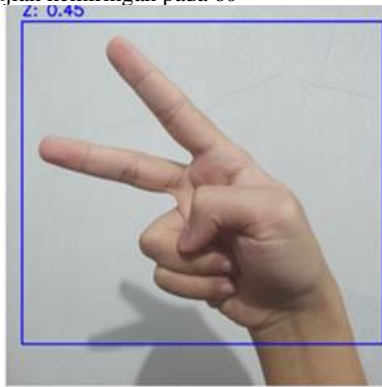


Gambar 22. Pengujian kemiringan pada 30° ke arah kanan

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", yang 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 milik kelas “V” dengan nilai kepercayaan 0,90.

- Pengujian kemiringan pada 60°



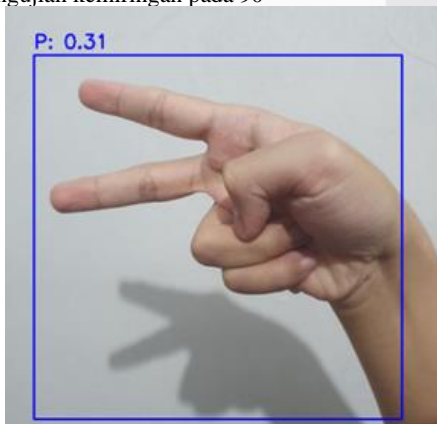
Gambar 23. Pengujian kemiringan pada 60° ke arah kiri



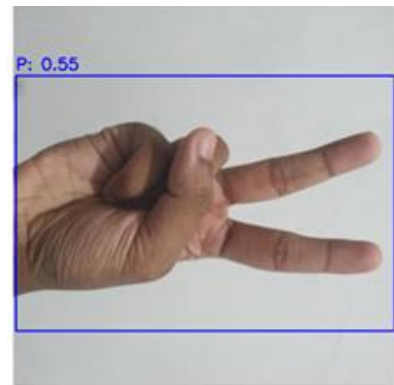
Gambar 24. Pengujian kemiringan pada 60° ke arah kanan

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

- 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 pada arah kiri maupun kanan, model YOLO mengidentifikasi gambar sebagai kelas “P”, yang merupakan kategorisasi yang salah. Seharusnya, gambar tersebut dikategorikan sebagai kelas “V”. Fenomena ini dapat dikaitkan dengan fakta bahwa ketika kelas “V” berada pada posisi kemiringan 90°, maka secara keliru dideteksi sebagai kelas “P”. Selain itu, kesamaan posisi antara kedua kelas tersebut dapat menyebabkan kebingungan dalam kemampuan model YOLOv8 untuk memprediksi kelas SIBI secara akurat.

V. KESIMPULAN

Dari penelitian yang sudah dilakukan didapatkan kesimpulan bahwa algoritma YOLOv8 dapat melakukan deteksi gambar pada bahasa isyarat SIBI. Algoritma YOLOv8 menunjukkan kinerja yang 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 di luar ruangan, bahkan dalam kondisi cahaya yang terang. Pada kondisi intensitas cahaya yang rendah, model YOLOv8 masih mampu melakukan prediksi mengenai kelas alfabet SIBI, meskipun akurasi nya tidak setinggi pada kondisi cahaya yang terang.

Pada kondisi di luar ruangan dengan cahaya redup, model YOLOv8 tidak mampu melakukan prediksi. Dalam pengujian jarak di mana kelas alfabet SIBI dapat dideteksi, 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° ke arah yang benar. Namun, model ini menunjukkan penurunan kinerja pada sudut yang melebihi kisaran ini, yang mengindikasikan adanya potensi keterbatasan dalam kemampuannya untuk secara akurat mendeteksi gambar dalam alfabet SIBI. Penting untuk diingat bahwa setiap alfabet dalam 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 secara akurat mengidentifikasi alfabet SIBI yang benar.

REFERENSI

- [1] J. Khikam Hikmalansya, D. Cahyono, dan S. Surabaya, “Aplikasi Pembelajaran Bahasa Isyarat Berbasis Android,” 2016.
- [2] Dian Naren, “Aksinya Viral saat Debat Pilpres 2024, Ternyata Segini Gaji Fantastis Juru Bicara Isyarat, Tertarik?” Diakses: 14 Agustus 2024. [Daring]. Tersedia pada: <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*, 2022, [Daring]. Tersedia pada: <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,” *2021 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, hlm. 28–33, 2021, [Daring]. Tersedia pada: <https://api.semanticscholar.org/CorpusID:246533726>
- [5] Dr. Samto, S. Pd. Dimiyati Hakim, Dra. Lani Bunawan, dan M. P. Dr. Totok Bintoro, “Kamus SIBI Kerjasama antara Kementerian Pendidikan dan Kebudayaan dengan Lembaga Penelitian dan Pengembangan Sistem Isyarat Bahasa Indonesia.”
- [6] M. Alaftekin, I. Pacal, dan K. Cicek, “Real-time sign language recognition based on YOLO algorithm,” *Neural Comput Appl*, vol. 36, no. 14, hlm. 7609–7624, Mei 2024, doi: 10.1007/s00521-024-09503-6.
- [7] ultralytics, “YOLOv8.” Diakses: 14 Agustus 2024. [Daring]. Tersedia pada: <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, hlm. 125122–125137, 2023, doi: 10.1109/ACCESS.2023.3330844.
- [9] “NVIDIA-SMI,” 2013. [Daring]. Tersedia pada: www.nvidia.com
- [10] Nisha Arya Ahmed, “What is A Confusion Matrix in Machine Learning? The Model Evaluation Tool Explained.” Diakses: 14 Agustus 2024. [Daring]. Tersedia pada: <https://www.datacamp.com/tutorial/what-is-a-confusion-matrix-in-machine-learning>
- [11] Jacob Solawetz, “What is Mean Average Precision (mAP) in Object Detection?,” Roboflow.