

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Justifikasi Solusi**

##### **2.1.1 IMAGE CLASSIFICATION – ULTRALYTICS YOLO DOCUMENTATION (YOLOv8-CLS)[7]**

Dokumentasi resmi Ultralytics menjelaskan bahwa image classification adalah tugas yang menghasilkan satu label kelas beserta confidence score untuk sebuah gambar, sehingga cocok digunakan ketika tujuan sistem hanya menentukan “gambar ini termasuk kelas apa”, tanpa perlu mengetahui lokasi objek seperti pada object detection. Penjelasan ini menjadi dasar pemilihan pendekatan pada penelitian ini karena keluaran yang dibutuhkan adalah huruf alfabet BISINDO (A–Z) dalam bentuk label tunggal, bukan bounding box. Selain itu, Ultralytics menyediakan model klasifikasi dengan penanda “-cls” (misalnya model pretrained untuk klasifikasi) yang mendukung penerapan transfer learning pada dataset khusus. Dengan demikian, penggunaan YOLO pada mode classification menjadi relevan dan konsisten dengan tujuan penelitian yang berfokus pada pengenalan kelas huruf dari citra gestur tangan.

##### **2.1.2 COMPARATIVE ANALYSIS OF CNN ARCHITECTURES FOR SIBI IMAGE CLASSIFICATION[8]**

Brianorman & Utami (2024) melakukan perbandingan beberapa arsitektur CNN pretrained (seperti VGG16, ResNet50, Inception, Xception, dan MobileNetV2) pada tugas klasifikasi citra bahasa isyarat SIBI. Studi ini memperkuat bahwa bahasa isyarat berbasis gestur dapat dimodelkan sebagai multi-class image classification, serta menunjukkan bahwa penggunaan model pretrained/transfer learning mampu memberikan performa tinggi pada klasifikasi gestur. Penelitian tersebut juga

menekankan bahwa evaluasi model klasifikasi tidak cukup hanya menggunakan akurasi, melainkan perlu dilengkapi dengan metrik precision, recall, dan F1-score agar performa per kelas dapat terlihat lebih jelas. Temuan dan pendekatan evaluasi ini menjadi acuan penelitian ini dalam menetapkan masalah sebagai klasifikasi alfabet BISINDO dan menggunakan metrik evaluasi klasifikasi secara lengkap.

### **2.1.3 INDONESIAN SIGN LANGUAGE ALPHABET IMAGE CLASSIFICATION USING VISION TRANSFORMER[9]**

Agustiansyah & Kurniadi (2025) meneliti klasifikasi alfabet BISINDO menggunakan pendekatan Vision Transformer dan menunjukkan bahwa alfabet BISINDO dapat dikenali melalui citra statis dengan memanfaatkan model deep learning yang kuat dalam mengekstraksi fitur visual. Studi ini menjadi penguat bahwa fokus penelitian pada level alfabet (A–Z) merupakan ruang yang valid dan relevan secara akademik, karena tantangan utama alfabet BISINDO memang terletak pada kemiripan bentuk gestur antar huruf yang bisa menyebabkan kekeliruan prediksi. Hal tersebut sejalan dengan penelitian ini yang melakukan evaluasi berbasis classification report dan confusion matrix untuk melihat huruf yang masih tertukar. Dengan demikian, penelitian ini mengambil acuan bahwa pendekatan image classification (tanpa bounding box) tetap tepat untuk pengenalan alfabet BISINDO, selama evaluasi dilakukan secara per kelas dan dianalisis kesalahan antar kelasnya.

### **2.1.4 INDONESIAN SIGN LANGUAGE (BISINDO) CLASSIFICATION USING K-FOLD CROSS-VALIDATION[10]**

Penelitian pada IJADIS (2025) membahas klasifikasi alfabet BISINDO dengan jumlah kelas 26 dan mengevaluasi performa model menggunakan K-Fold cross-validation, sekaligus melaporkan metrik seperti F1-score sebagai indikator performa yang lebih seimbang dalam kasus klasifikasi

multi-kelas. Studi ini menguatkan bahwa pengenalan alfabet BISINDO memang lazim diposisikan sebagai multi-class classification, serta penting untuk menilai performa secara lebih detail (bukan hanya akurasi global) karena tiap huruf dapat memiliki tingkat kesulitan yang berbeda. Oleh karena itu, penelitian ini menjadikan penelitian tersebut sebagai acuan dalam menekankan penggunaan metrik precision–recall–F1 dan analisis kesalahan antar kelas melalui confusion matrix, sehingga evaluasi yang dihasilkan tidak sekadar menyatakan “akurasi tinggi”, tetapi juga menjelaskan kelas mana yang masih perlu perbaikan.

## 2.2 Tinjauan Teori

### 2.2.1 Bahasa Isyarat Indonesia (BISINDO)

Bahasa Isyarat Indonesia (BISINDO) adalah ragam bahasa isyarat yang digunakan komunitas Tuli di Indonesia, dengan variasi lokal yang kuat antarkota (mis. Jakarta vs Yogyakarta) dan sejarah kontak dengan keluarga bahasa isyarat Prancis/ASL. Temuan sosiolinguistik menunjukkan BISINDO bukanlah satu sistem tunggal yang sepenuhnya seragam; varietasnya bisa berbeda pada leksikon dan tata bahasa. Contohnya, urutan kata cenderung SVO di Jakarta dan SOV di Yogyakarta sehingga pendekatan komputasional perlu peka terhadap keragaman bentuk dan distribusi visual gerak isyarat. Dalam konteks pendidikan formal Indonesia, SIBI (Sistem Isyarat Bahasa Indonesia) juga digunakan sebagai bentuk isyarat terkodekan dari bahasa Indonesia, berbeda dari BISINDO yang tumbuh secara alami di komunitas Tuli. Perbedaan konseptual ini penting ketika menyiapkan data dan menyatakan klaim generalisasi model[11].

Kajian akademik menegaskan koeksistensi BISINDO dan SIBI di Indonesia serta menyoroti tantangan rekognisi otomatis akibat variasi bentuk, latar budaya, dan setting komunikasi (percakapan natural vs alfabet manual). Untuk peneliti visi komputer, variasi intra-kelas (mis. variasi individu, sudut kamera, pencahayaan, latar) dan antar-varietas (Jakarta/Yogyakarta) berdampak pada distribusi data dan metrik evaluasi.

Karena itu, desain dataset dan protokol validasi harus mempertimbangkan representasi variasi yang memadai (person split, background diversity) agar model deteksi/pengenalan handshape dan gerakan dapat teruji secara lebih robust[12].

### 2.2.2 Ultralytics

Ultralytics adalah organisasi di balik pengembangan YOLO modern (v5/v8) beserta ekosistem yang memudahkan pelatihan, validasi, dan deployment. Paket ultralytics menyediakan API Python dan CLI yang konsisten lintas tugas (deteksi, segmentasi, klasifikasi, pose, OBB), sehingga eksperimen model bisa direproduksi dengan sedikit baris kode dan konfigurasi yang eksplisit (mis. data.yaml, imgsz, epochs, augmentasi). Dokumentasi resmi menyediakan panduan end-to-end (pengaturan dataset, training model.train(), validasi model.val(), hingga ekspor model). Dari perspektif rekayasa, integrasi tooling Ultralytics mempermudah siklus eksperimen: logging metrik (mAP, P/R, F1), resume training, dan ekspor ke berbagai backend (ONNX, TensorRT, CoreML) untuk keperluan real-time. Abstraksi ini mengurangi “overhead” implementasi, sehingga fokus dapat bergeser ke kurasi dataset BISINDO yang baik, pemilihan ukuran model (n/s/m/l/x), serta strategi augmentasi untuk meningkatkan generalisasi pada kondisi dunia nyata[13].

### 2.2.3 YOLOv8

YOLOv8 adalah evolusi terbaru dari algoritma deteksi objek You Only Look Once (YOLO), yang dikembangkan oleh Ultralytics. Model ini dirancang untuk menawarkan kinerja deteksi objek yang sangat baik dengan kecepatan inferensi yang tinggi dan kemampuan untuk mendeteksi objek dalam berbagai kondisi pencahayaan dan sudut pandang[14]. Arsitektur YOLOv8 terdiri dari beberapa komponen utama yang bekerja bersama untuk mencapai performa deteksi yang optimal:

## 1. Backbone

YOLOv8 menggunakan CSPDarknet53 sebagai backbone, yang merupakan varian dari Darknet. CSP (Cross-Stage Partial) membantu meningkatkan efisiensi pelatihan dan mengurangi kebutuhan komputasi dengan membagi representasi fitur menjadi dua bagian, yang memungkinkan model untuk mempercepat proses pembelajaran tanpa kehilangan detail penting.

## 2. Neck

YOLOv8 mengadopsi PANet (Path Aggregation Network) pada bagian neck-nya. PANet digunakan untuk menggabungkan fitur multi-skala secara efektif, yang memungkinkan model untuk mendeteksi objek dengan berbagai ukuran. Teknik ini membantu YOLOv8 dalam menangani objek yang kecil maupun besar dengan lebih baik dibandingkan dengan versi sebelumnya.

## 3. Head

Pada bagian head, YOLOv8 melakukan deteksi objek dengan menggunakan lapisan konvolusional untuk menghasilkan output berupa kelas objek, koordinat bounding box, dan skor keyakinan. YOLOv8 menggunakan metode anchor-free, yang memudahkan model untuk mendeteksi objek tanpa ketergantungan pada bounding box yang sudah ditentukan sebelumnya (anchor boxes).

## 4. Loss Function

YOLOv8 menggunakan kombinasi loss functions untuk mengoptimalkan performa deteksi, yang mencakup:

- Bounding box loss: Mengukur kesalahan dalam penempatan dan ukuran bounding box.
- Confidence loss: Mengukur ketepatan skor keyakinan pada objek yang terdeteksi.

- Classification loss: Mengukur ketepatan dalam klasifikasi objek yang terdeteksi.

## 5. Optimisasi dan Fitur Lain

YOLOv8 mengintegrasikan Self-Attention (SA) untuk memperbaiki fokus model pada objek kecil yang sulit dideteksi. Fitur lain yang penting adalah Fused-Multiply-Add (FMA), yang meningkatkan kecepatan pemrosesan tanpa mengurangi kualitas deteksi. Dengan fitur-fitur ini, YOLOv8 mampu melakukan inferensi yang cepat namun tetap mempertahankan tingkat akurasi yang tinggi.

### 2.2.4 YOLOv8 untuk Image Classification

YOLOv8 merupakan keluarga model dari Ultralytics yang mendukung beberapa tugas visi komputer. Selain dikenal pada ranah deteksi, YOLOv8 juga menyediakan mode image classification yang ditandai dengan model berakhiran “-cls” (misalnya yolov8s-cls.pt). Pada mode klasifikasi, model menghasilkan prediksi kelas dan confidence score, sehingga cocok untuk kebutuhan penelitian yang ingin mengenali huruf dari gambar gestur tangan tanpa bounding box[7].

Dalam praktik pelatihan, YOLOv8 classification umumnya memanfaatkan model pretrained sehingga proses pembelajaran dapat dilakukan lebih efisien melalui *transfer learning*, terutama ketika dataset yang digunakan tidak terlalu besar. Penggunaan model pretrained membantu model memiliki pengetahuan awal mengenai pola visual umum dan kemudian disesuaikan dengan karakteristik data alfabet BISINDO.