

Komparasi Model YOLO untuk Sistem Penerjemah Gestur BISINDO Real-Time Berbasis ESP32-CAM

Daffin Tanjiro Widison^{*1}, Dwi Arman Prasetya², Alfian Rizaldy Pratama³

^{1, 2, 3}Departemen Sains Data, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia
Email: ¹22083010010@student.upnjatim.ac.id, ²arman.prasetya.sada@upnjatim.ac.id,
³alfan.fasilkom@upnjatim.ac.id

Abstrak

Bahasa Isyarat Indonesia (BISINDO) merupakan media komunikasi bagi penyandang tunarungu dan tunawicara, namun keterbatasan pemahaman bahasa isyarat oleh masyarakat umum masih menjadi kendala dalam komunikasi. Salah satu tantangan dalam pengembangan sistem penerjemah BISINDO berbasis *Internet of Things* (IoT) adalah menyeimbangkan akurasi deteksi dan latensi agar sistem dapat bekerja secara *real-time* pada perangkat dengan sumber daya terbatas. Penelitian ini mengembangkan sistem penerjemah gestur BISINDO menggunakan ESP32-CAM dan metode deteksi objek YOLO dengan keluaran berupa teks dan suara. Penelitian membandingkan lima model, yaitu YOLOv8m, YOLOv9m, YOLOv10m, YOLO11m, dan YOLO26m, menggunakan *dataset* gabungan primer dan sekunder sebanyak 7.169 citra yang terdiri atas alfabet dan 30 kata dasar BISINDO. Evaluasi model dilakukan menggunakan *precision*, *recall*, *mAP50*, *mAP50-95*, *F1-score*, dan latensi pada lingkungan pengujian NVIDIA GeForce GTX 1050. Hasil pengujian menunjukkan bahwa YOLOv8m memberikan keseimbangan performa terbaik dengan nilai *mAP50-95* sebesar 81,3% dan latensi sebesar 53,91 *ms/image*. Pada implementasi sistem, diperoleh *accuracy* sebesar 96,8%, *precision* 97,8%, *recall* 96,8%, dan *F1-score* 96,6%, dengan latensi komunikasi sebesar 47–52 ms. Hasil tersebut menunjukkan bahwa model YOLOv8m mampu memberikan performa deteksi yang akurat dengan latensi yang responsif untuk mendukung penerjemahan gestur BISINDO secara *real-time* menggunakan ESP32-CAM.

Kata kunci: BISINDO, deteksi objek, *Internet of Things* (IoT), penerjemah bahasa isyarat, YOLO

Abstract

Indonesian Sign Language (BISINDO) serves as a means of communication for deaf and speech-impaired individuals; however, limited public understanding of sign language remains a challenge in communication. One of the technical challenges in developing an *Internet of Things* (IoT)-based BISINDO translation system is balancing detection accuracy and latency to enable *real-time* operation on resource-constrained devices. This study develops a BISINDO gesture translation system using the ESP32-CAM and the YOLO object detection method, with text and audio outputs. Five models, namely YOLOv8m, YOLOv9m, YOLOv10m, YOLO11m, and YOLO26m, were compared using a combined primary and secondary dataset consisting of 7,169 images covering the BISINDO alphabet and 30 basic words. Model performance was evaluated using *precision*, *recall*, *mAP50*, *mAP50-95*, *F1-score*, and latency on an NVIDIA GeForce GTX 1050 testing environment. The results show that YOLOv8m provides the best performance balance, achieving an *mAP50-95* of 81.3% with a latency of 53.91 *ms/image*. In system implementation, the system achieved an *accuracy* of 96.77%, *precision* of 97.82%, *recall* of 96.77%, and *F1-score* of 96.63%, with communication latency ranging from 47 to 52 ms. These results demonstrate that the YOLOv8m model provides accurate detection performance with responsive latency, supporting *real-time* BISINDO gesture translation using the ESP32-CAM.

Keywords: BISINDO, *Internet of Things* (IoT), object detection, sign language translator, YOLO

This work is an open access article and licensed under a Creative Commons Attribution-NonCommercial ShareAlike 4.0 International (CC BY-NC-SA 4.0)



1. PENDAHULUAN

Penyandang disabilitas merupakan kelompok masyarakat yang menghadapi keterbatasan dalam aspek fisik, mental, intelektual, maupun sensorik. Kondisi ini mengakibatkan adanya kesulitan dalam berinteraksi dengan lingkungan sekitar. Salah satu bentuk disabilitas yang sering menimbulkan hambatan komunikasi adalah tuna rungu dan tuna wicara. Berdasarkan data yang tercatat hingga tahun 2009, terdapat sebanyak 2.126.000 jiwa penyandang disabilitas di Indonesia [1]. Dari jumlah tersebut,

223.665 jiwa merupakan penyandang tuna rungu, 151.371 jiwa penyandang tuna wicara, dan 73.560 jiwa menyandang tuna rungu wicara. Secara keseluruhan, sekitar 21% dari total penyandang disabilitas di 24 provinsi di Indonesia mengalami disabilitas rungu wicara [1].

Bahasa isyarat merupakan bentuk bahasa yang memanfaatkan gerakan bibir, tubuh, dan tangan untuk menyampaikan maksud dalam komunikasi. Bagi penyandang tuna rungu dan tuna wicara, bahasa isyarat menjadi sarana komunikasi utama yang memungkinkan mereka menyampaikan pesan dan berinteraksi dalam kehidupan sehari-hari. Namun, sebagian besar masyarakat Indonesia belum memahami bahasa isyarat dan cenderung enggan untuk mempelajarinya. Hal ini menimbulkan hambatan yang signifikan ketika penyandang disabilitas harus berinteraksi dengan masyarakat umum. Kondisi tersebut tidak hanya menciptakan keterbatasan komunikasi, tetapi juga memperlebar kesenjangan dalam berbagai aspek kehidupan, seperti pendidikan, pekerjaan, hingga akses terhadap layanan publik [2].

Perkembangan visi komputer pada awalnya berfokus pada tugas klasifikasi citra, yaitu mengenali jenis objek tanpa mengetahui posisi keberadaannya dalam gambar. Seiring meningkatnya kebutuhan aplikasi dunia nyata, pendekatan tersebut berkembang menjadi deteksi objek yang mampu mengidentifikasi kelas sekaligus lokasi objek secara akurat [3]. Dalam dua dekade terakhir, deteksi objek menjadi salah satu fokus utama dalam bidang visi komputer. Pada tahap awal, metode yang digunakan masih berbasis *machine learning* konvensional dengan ekstraksi fitur secara manual [4]. Meskipun relatif ringan, pendekatan ini memiliki keterbatasan karena sangat bergantung pada kondisi pencahayaan, latar belakang, dan variasi sudut pandang objek. Perkembangan *deep learning* kemudian membawa peningkatan signifikan dalam performa deteksi objek, khususnya melalui arsitektur berbasis *Convolutional Neural Network* (CNN) yang mampu melakukan ekstraksi fitur secara otomatis dan lebih *robust* terhadap variasi kondisi lingkungan [5].

Salah satu arsitektur deteksi objek yang banyak digunakan adalah YOLO (*You Only Look Once*) sebagai metode deteksi objek *real-time* yang memformulasikan deteksi objek sebagai permasalahan regresi dengan memprediksi *bounding box* dan probabilitas kelas secara langsung dari keseluruhan citra dalam satu proses evaluasi. Pendekatan tersebut memungkinkan YOLOv1 mencapai kecepatan deteksi hingga 45 FPS dengan performa yang lebih baik dibandingkan metode tradisional seperti R-CNN dan DPM [6]. Seiring perkembangannya, arsitektur YOLO terus mengalami peningkatan, di mana YOLOv2 memperkenalkan penggunaan *anchor box*, YOLOv3 meningkatkan akurasi melalui *multi-scale feature fusion*, dan YOLOv8 mengadopsi mekanisme *anchor-free detection* untuk meningkatkan efisiensi serta performa deteksi pada aplikasi *real-time* [7].

Berbagai penelitian terdahulu telah memanfaatkan teknik *deep learning* untuk mengembangkan sistem deteksi dan penerjemahan Bahasa Isyarat Indonesia (BISINDO). Salah satu penelitian mengimplementasikan arsitektur *MobileNet V2 FPNLite SSD* menggunakan *TensorFlow Object Detection API* untuk mengenali 26 kelas huruf BISINDO, dengan rata-rata presisi 0,712 dan *recall* 0,747 serta mampu berjalan secara *real-time* pada 30 *frames per second* [8]. Penelitian lain menggunakan algoritma YOLOv8 untuk mendeteksi gestur BISINDO dan menunjukkan performa yang sangat tinggi dengan akurasi 95,8%, *recall* 97,4%, serta mAP50 sebesar 99,5% [9]. Hasil tersebut menunjukkan bahwa model berbasis YOLO memiliki keunggulan dalam hal kecepatan dan akurasi, sehingga sangat potensial untuk diterapkan pada sistem deteksi bahasa isyarat secara *real-time*.

Selain berbasis perangkat lunak, pengembangan sistem penerjemah bahasa isyarat juga telah dilakukan melalui integrasi dengan perangkat *Internet of Things* (IoT). Sebuah penelitian mengembangkan sarung tangan penerjemah berbasis sepuluh sensor *flex* yang terhubung ke mikrokontroler dan menghasilkan keluaran suara dengan tingkat keberhasilan 86,67% pada abjad A–G, meskipun performanya menurun saat terjadi transisi gerakan [10]. Penelitian lain memanfaatkan Raspberry Pi dengan metode *Blazepose* untuk mendeteksi huruf, angka, dan kata dengan akurasi serta kecepatan yang optimal untuk pengguna tertentu [11]. Selain itu, sistem berbasis kamera *real-time* dengan pendekatan CNN-LSTM yang terintegrasi dengan ESP32 mampu mencapai akurasi 91,4% dengan latensi rata-rata 3,1 detik dan tingkat keberhasilan keluaran suara sebesar 86,7% [12]. Meskipun demikian, sebagian besar penelitian tersebut masih memiliki keterbatasan dalam hal latensi, kompleksitas perangkat, serta belum optimal dalam menggabungkan performa deteksi yang tinggi dengan efisiensi komputasi pada perangkat terbatas.

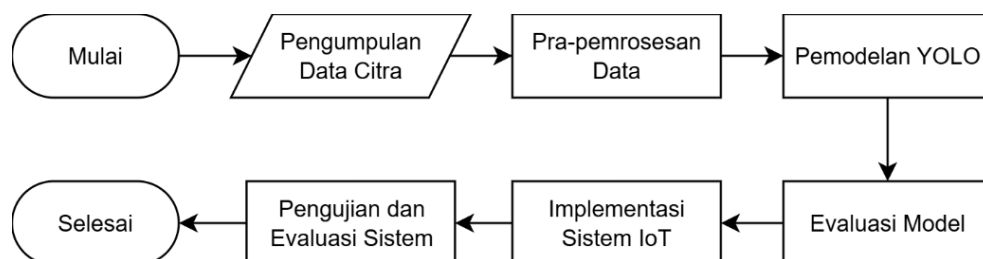
Berdasarkan penelitian tersebut, pengembangan sistem penerjemah BISINDO telah mencakup berbagai pendekatan, mulai dari deteksi objek berbasis *deep learning* hingga pemanfaatan sensor dan

integrasi perangkat IoT. Namun, setiap pendekatan memiliki karakteristik dan keterbatasan yang berbeda. Sistem berbasis sensor memerlukan perangkat tambahan yang harus digunakan oleh pemakai, sedangkan sistem berbasis kamera menghadapi tantangan dalam menyeimbangkan akurasi deteksi dan kecepatan pemrosesan. Penelitian yang menggunakan YOLOv8 telah menunjukkan hasil deteksi yang tinggi [9], tetapi masih diperlukan evaluasi komparatif terhadap beberapa varian YOLO yang lebih baru menggunakan *dataset* dan lingkungan pengujian yang sama. Selain itu, studi yang menggabungkan perbandingan beberapa varian YOLO tersebut dengan implementasi pada sistem IoT berdaya rendah serta menyediakan keluaran teks dan suara masih terbatas. ESP32-CAM dipilih sebagai perangkat akuisisi citra karena mengintegrasikan kamera dan konektivitas nirkabel dalam modul yang ringkas, relatif hemat daya, dan berbiaya rendah, sehingga sesuai untuk pengembangan prototipe IoT dengan keterbatasan sumber daya [13]. Dengan demikian, celah penelitian terletak pada perlunya evaluasi menyeluruh terhadap *trade-off* antara performa deteksi dan latensi inferensi sekaligus penerapannya dalam sistem penerjemah BISINDO yang terintegrasi.

Berdasarkan celah penelitian tersebut, penelitian ini bertujuan untuk mengembangkan prototipe sistem penerjemah BISINDO berbasis IoT menggunakan ESP32-CAM sebagai perangkat akuisisi citra dan server lokal untuk pemrosesan inferensi, dengan keluaran berupa teks dan suara. Selain itu, penelitian ini bertujuan membandingkan kinerja lima model deteksi objek, yaitu YOLOv8m, YOLOv9m, YOLOv10m, YOLO11m, dan YOLO26m, menggunakan *dataset* BISINDO dan lingkungan pengujian yang sama. Evaluasi dilakukan berdasarkan metrik *precision*, *recall*, mAP50, mAP50-95, F1-score, dan latensi untuk menganalisis keseimbangan antara akurasi deteksi dan kecepatan inferensi. Kontribusi penelitian ini meliputi pengembangan sistem penerjemah BISINDO yang mengintegrasikan akuisisi citra, pemrosesan berbasis YOLO, keluaran teks dan suara, evaluasi komparatif lima varian YOLO pada skenario pengujian yang seragam, dan analisis *trade-off* antara kinerja deteksi dan latensi sebagai dasar pemilihan model yang sesuai untuk implementasi penerjemah BISINDO berbasis IoT dengan pemrosesan pada server lokal.

2. METODE PENELITIAN

Penelitian ini dilakukan melalui serangkaian tahapan yang mencakup perancangan arsitektur sistem berbasis *Internet of Things* (IoT), pengumpulan dan pengolahan *dataset* gestur BISINDO, pemilihan serta pelatihan model deteksi objek YOLO, hingga evaluasi kinerja sistem secara menyeluruh. Pendekatan yang digunakan berfokus pada penerapan metode deteksi objek berbasis *deep learning* untuk mengenali gestur tangan BISINDO secara *real-time*, disertai analisis komparatif terhadap beberapa varian model guna memperoleh performa yang optimal. Dalam implementasinya, sistem mengintegrasikan perangkat ESP32-CAM dengan lingkungan pemrograman Python sebagai pusat pemrosesan, sehingga memungkinkan pengolahan data citra, inferensi model, serta komunikasi data berlangsung secara efisien melalui jaringan IoT. Selain itu, penelitian ini juga mempertimbangkan aspek implementasi pada perangkat keras serta skenario pengujian yang dirancang untuk mengevaluasi akurasi, kecepatan, dan efisiensi sistem dalam kondisi penggunaan nyata. Penelitian ini disusun dengan menggunakan alur yang ditunjukkan pada Gambar 1.



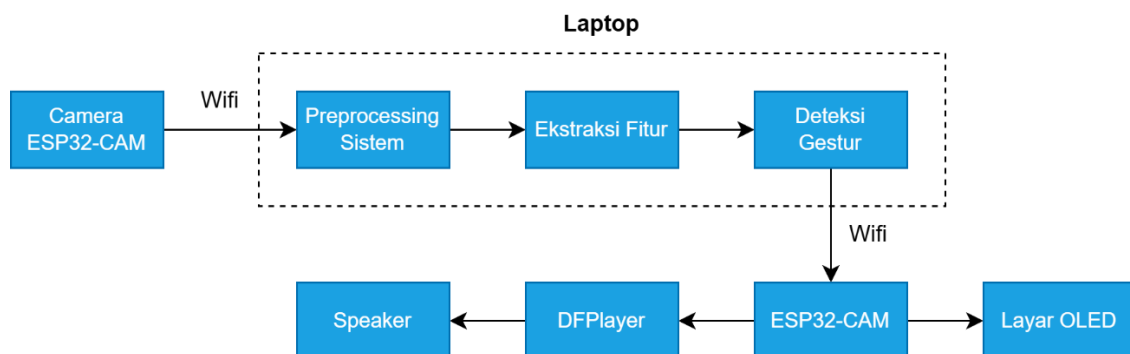
Gambar 1. Diagram Alir Penelitian

Berdasarkan diagram alir penelitian pada Gambar 1, tahap pertama dalam penelitian ini adalah pengumpulan data citra yang dilakukan untuk memperoleh *dataset* gestur BISINDO dari sumber primer dan sekunder. Selanjutnya, data memasuki tahap pra-pemrosesan yang mencakup penyesuaian ukuran citra, anotasi, augmentasi data, serta pembagian *dataset* menjadi data latih, validasi, dan pengujian. Tahap berikutnya adalah pemodelan data dengan melakukan pelatihan beberapa varian model YOLO

untuk mendeteksi gestur BISINDO secara *real-time*. Model yang telah dilatih kemudian dievaluasi menggunakan berbagai metrik performa guna menentukan model terbaik berdasarkan tingkat akurasi dan efisiensi inferensi. Setelah model optimal diperoleh, tahap implementasi sistem IoT dilakukan dengan mengintegrasikan model deteksi ke dalam sistem berbasis ESP32-CAM beserta perangkat keluaran seperti OLED dan *speaker*. Tahap terakhir adalah pengujian dan evaluasi sistem untuk menilai performa keseluruhan sistem dalam kondisi penggunaan nyata, termasuk pengujian deteksi gestur, keluaran sistem, serta pengukuran latensi komunikasi dan pemrosesan data.

2.1. Arsitektur Sistem

Penelitian ini menggunakan rancangan arsitektur sistem berbasis *Internet of Things* (IoT) dengan memanfaatkan ESP32-CAM sebagai mikrokontroler utama. Perangkat ini dipilih karena memiliki modul kamera terintegrasi, kemampuan konektivitas Wi-Fi, serta ukuran yang ringkas dan konsumsi daya yang relatif rendah, sehingga sesuai untuk implementasi sistem portabel dan *real-time* [13]. Selain itu, sistem juga memanfaatkan DFPlayer sebagai modul pemutar audio untuk mengatur luaran suara, yang memiliki keunggulan berupa kemudahan integrasi, dukungan penyimpanan *file* suara pada media eksternal, serta kemampuan menghasilkan *output* audio secara mandiri tanpa memerlukan pemrosesan tambahan yang kompleks [14]. Untuk menampilkan hasil deteksi, digunakan layar OLED yang memiliki kelebihan pada konsumsi daya yang rendah, kontras tampilan yang tinggi, serta kemampuan menampilkan teks dengan jelas meskipun pada ukuran layar yang kecil [15]. Arsitektur sistem yang dibangun dalam penelitian ini ditunjukkan pada blok diagram Gambar 2.



Gambar 2. Blok Diagram Arsitektur Sistem

Berdasarkan diagram arsitektur sistem yang ditunjukkan pada Gambar 2, proses diawali dengan akuisisi data berupa citra gestur tangan menggunakan kamera pada ESP32-CAM. Data yang diperoleh kemudian dikirimkan melalui jaringan Wi-Fi menuju server lokal untuk diproses lebih lanjut. Pada tahap pemrosesan, model deteksi objek berbasis YOLO digunakan untuk mengidentifikasi dan mengklasifikasikan gestur tangan menjadi representasi teks. Hasil deteksi tersebut selanjutnya diteruskan sebagai keluaran sistem dalam bentuk tampilan teks pada layar serta suara melalui perangkat audio, sehingga dapat digunakan sebagai media komunikasi antara pengguna BISINDO dan masyarakat umum. Seluruh rangkaian proses ini dirancang untuk berjalan secara *real-time* dengan mempertimbangkan efisiensi komputasi dan keterbatasan sumber daya perangkat.

2.2. Spesifikasi Server Lokal

Untuk menjamin reproduksibilitas eksperimen, proses pelatihan dan evaluasi model dilakukan pada lingkungan komputasi dengan spesifikasi perangkat keras dan perangkat lunak yang didokumentasikan secara terperinci. Rincian spesifikasi lingkungan komputasi dan konfigurasi pelatihan disajikan pada Tabel 1.

Tabel 1. Spesifikasi Server Lokal

Komponen	Spesifikasi
CPU	AMD Ryzen 5 3550H
RAM	16 GB DDR4
GPU	NVIDIA GeForce GTX 1050
VRAM	3 GB
Sistem operasi	Windows 11 64-bit
Python	3.12.7
Ultralytics	8.4.52
CUDA	13.0

Spesifikasi lingkungan eksperimen dan konfigurasi perangkat lunak dirangkum pada tabel untuk memberikan gambaran mengenai lingkungan komputasi yang digunakan selama proses pelatihan dan evaluasi model.

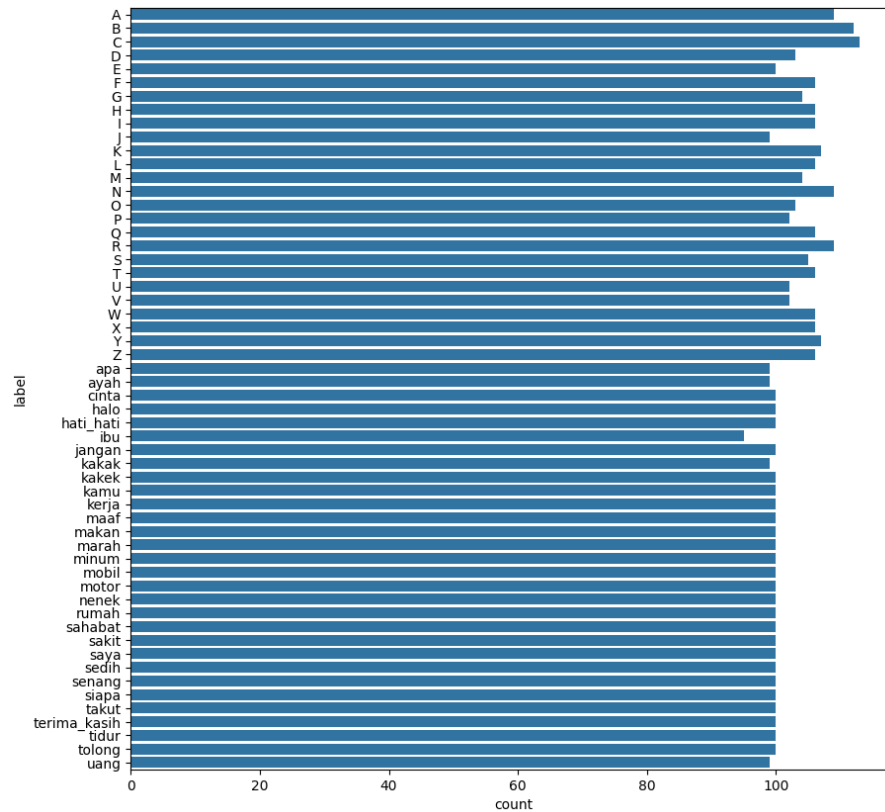
2.3. Sumber Data

Penelitian ini menggunakan kombinasi antara *dataset* primer dan *dataset* sekunder untuk meningkatkan keberagaman serta kualitas data. *Dataset* primer diperoleh melalui proses pengambilan data secara langsung oleh penulis menggunakan kamera, dengan jumlah sebanyak 2.020 citra gestur BISINDO. Pengambilan data dilakukan dengan mempertimbangkan variasi kondisi, meliputi dua orang peraga, lima jenis latar belakang, tiga jenis pakaian, serta variasi posisi dan letak tangan guna meningkatkan kemampuan generalisasi model. Selain itu, penelitian ini juga memanfaatkan *dataset* sekunder yang diperoleh dari platform Roboflow dengan jumlah sebanyak 5.149 citra [16], [17].

Gestur BISINDO yang digunakan dalam kedua dataset mencakup 26 kelas huruf alfabet dan 30 kelas kata dasar, sehingga total terdapat 56 kelas yang digunakan dalam proses pelatihan dan evaluasi model. Cakupan kelas tersebut bertujuan mendukung kebutuhan komunikasi dasar dan proses pembelajaran bahasa isyarat bagi penyandang disabilitas [18]. Seluruh data kemudian melalui proses anotasi menggunakan perangkat lunak LabelMe berupa pemberian *bounding box* dan label kelas pada setiap citra gestur BISINDO [19]. Proses pelabelan dilakukan secara manual oleh salah satu penulis untuk menjaga konsistensi anotasi antar data. Setiap gambar diberikan satu anotasi utama yang merepresentasikan satu kelas gestur sesuai objek tangan yang ditampilkan pada citra.

2.4. Pembagian Data Validasi

Tahap pembagian data dilakukan secara acak menjadi tiga *subset*, yaitu data latih sebesar 80% (5.735 citra), data validasi sebesar 10% (717 citra), dan data uji sebesar 10% (717 citra), dari total 7.169 citra yang telah melalui tahap *data cleaning* dan transformasi. Data latih terdiri atas 4.136 citra data sekunder dan 1.599 citra data primer, sedangkan data validasi terdiri atas 515 citra data sekunder dan 202 citra data primer. Adapun data uji terdiri atas 498 citra data sekunder dan 219 citra data primer. Untuk meningkatkan kemampuan generalisasi model, augmentasi diterapkan hanya pada data latih melalui rotasi citra hingga 10°, horizontal *flipping* dengan probabilitas 50%, serta penyesuaian *brightness* dan *contrast* sebesar 20%, sehingga jumlah data latih meningkat dua kali lipat. Sementara itu, data validasi dan data uji tidak diaugmentasi agar evaluasi dilakukan menggunakan citra yang tidak dimodifikasi, sehingga hasil pengukuran performa lebih objektif dalam merepresentasikan kemampuan model pada data yang tidak digunakan dalam pelatihan. Distribusi jumlah citra untuk setiap kelas pada data latih ditunjukkan pada Gambar 3.



Gambar 3. Distribusi Kelas pada Data Latih

Berdasarkan hasil visualisasi tersebut, distribusi data antar kelas relatif seimbang dengan jumlah sampel yang tidak berbeda secara signifikan. Kelas dengan jumlah data paling sedikit adalah kelas *ibu* sebanyak 119 citra, sedangkan kelas dengan jumlah data terbanyak adalah kelas *C* sebanyak 141 citra. Selisih jumlah data antara kelas terendah dan tertinggi hanya sebesar 22 citra atau sekitar 15,6% dari kelas terbesar, sehingga distribusi data tidak menunjukkan karakteristik *class imbalance* yang ekstrem.

2.5. Pelatihan dan Pemilihan Model Terbaik

Proses pelatihan dilakukan dengan membandingkan lima varian model deteksi objek, yaitu YOLOv8m, YOLOv9m, YOLOv10m, YOLO11m, dan YOLO26m, guna menganalisis perbedaan performa dalam mendeteksi gestur BISINDO. Penamaan YOLO26 merujuk pada model YOLO yang dirilis pada tahun 2026, sedangkan akhiran m menunjukkan varian ukuran medium [20]. Parameter pelatihan yang digunakan pada seluruh model ditunjukkan pada Tabel 2.

Tabel 2. <i>Hyperparameter</i> Pelatihan Model	
<i>Hyperparameter</i>	Nilai
<i>Epoch</i>	30
<i>Image Size</i>	640 × 640
<i>Batch Size</i>	16
<i>Optimizer</i>	<i>Auto Optimizer</i> (SGD/AdamW)
<i>Initial Learning Rate</i>	0.01

Berdasarkan *hyperparameter* pelatihan model yang digunakan, seluruh model dilatih menggunakan ukuran citra 640×640 piksel dengan *batch size* 16 selama 30 *epoch*. Proses optimasi dilakukan menggunakan *auto optimizer* bawaan *framework* Ultralytics YOLO yang secara otomatis menyesuaikan jenis *optimizer*, seperti SGD atau AdamW, berdasarkan karakteristik model dan *dataset* [21]. Selain itu, *initial learning rate* ditetapkan sebesar 0,01 untuk menjaga stabilitas proses pembelajaran selama pelatihan. Model YOLO12m tidak disertakan dalam penelitian ini karena memiliki kebutuhan

komputasi yang relatif tinggi sehingga memerlukan waktu pelatihan lebih lama dan kurang efisien untuk skenario penelitian yang berfokus pada implementasi sistem *real-time* [22].

Pemilihan model terbaik dilakukan berdasarkan hasil evaluasi kinerja menggunakan metrik *mean Average Precision* pada rentang IoU 0.5 hingga 0.95 (mAP50–95) untuk mengukur tingkat akurasi deteksi secara menyeluruh, serta latensi untuk menilai kecepatan inferensi model [23]. Kombinasi kedua metrik ini digunakan sebagai dasar pertimbangan dalam menentukan model yang paling optimal, dengan memperhatikan keseimbangan antara akurasi dan efisiensi waktu pemrosesan sehingga sesuai untuk diimplementasikan pada sistem penerjemah BISINDO berbasis IoT secara *real-time*.

2.6. Pengujian dan Evaluasi Sistem

Pengujian dan evaluasi sistem dilakukan untuk menilai kinerja model serta keandalan integrasi perangkat dalam kondisi penggunaan nyata. Proses pengujian diawali dengan perekaman video seluruh gestur BISINDO menggunakan perangkat IoT berbasis ESP32-CAM dengan durasi masing-masing selama 5 detik. Rekaman video tersebut kemudian dikirim ke server lokal untuk dilakukan proses deteksi gestur. Sebelum tahap deteksi, video diolah dengan cara mengekstraksi *frame* menjadi 10 citra per detik, sehingga setiap *frame* dapat dianalisis secara individual oleh model deteksi objek untuk mengidentifikasi jenis gestur yang ditampilkan. Hasil deteksi dari setiap *frame* selanjutnya dievaluasi menggunakan *confusion matrix* untuk mengukur kinerja model secara komprehensif, yang meliputi metrik akurasi, presisi, *recall*, dan *F1-score* [24].

Selain itu, dilakukan pengukuran latensi untuk mengetahui waktu yang dibutuhkan dalam proses komunikasi dan pemrosesan data antara ESP32-CAM dan server lokal yang menjalankan model YOLO. Pengujian dilakukan dengan mengirimkan citra beresolusi 640×480 piksel menggunakan protokol HTTP melalui jaringan *hotspot* perangkat Android yang berfungsi sebagai media komunikasi antara perangkat IoT dan server deteksi. Selama proses pengujian, sistem dijalankan selama 30 detik dengan melakukan deteksi gestur BISINDO secara *real-time* untuk merekam waktu respons komunikasi dan inferensi pada setiap proses pengiriman data. Hasil pengukuran tersebut digunakan untuk mengevaluasi tingkat responsivitas dan kestabilan performa sistem dalam kondisi penggunaan nyata.

3. HASIL

3.1. Hasil Perancangan Blok Diagram Sistem

Hasil perancangan blok diagram sistem menunjukkan integrasi antara perangkat keras dan perangkat lunak dalam membangun sistem penerjemah Bahasa Isyarat Indonesia (BISINDO) berbasis *Internet of Things* (IoT). Sistem dirancang untuk mendukung proses deteksi gestur secara *real-time* dengan memanfaatkan ESP32-CAM sebagai perangkat akuisisi citra, model YOLO sebagai metode deteksi objek, serta perangkat keluaran berupa layar OLED dan *speaker* untuk menyampaikan hasil penerjemahan dalam bentuk teks dan suara. Gambar 4 menunjukkan prototipe sistem penerjemah BISINDO yang telah dikembangkan pada penelitian ini.



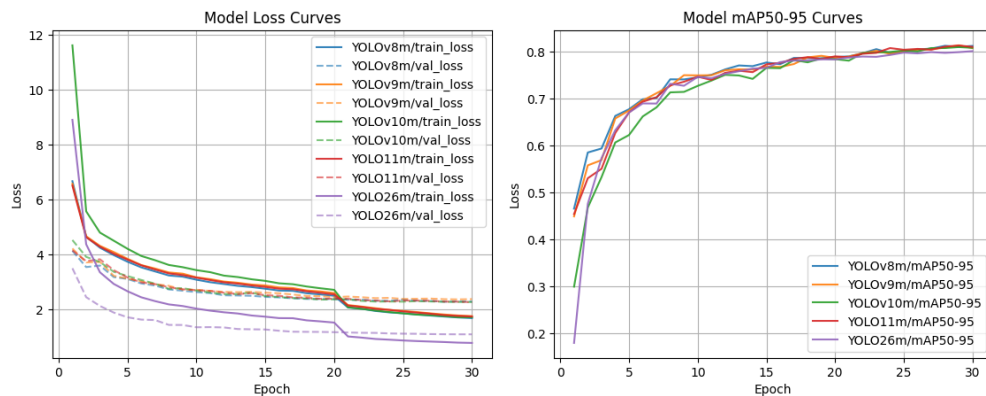
Gambar 4. Prototipe Penerjemah BISINDO

Berdasarkan Gambar 4, prototipe sistem penerjemah BISINDO dirancang dalam bentuk perangkat portabel dengan dimensi panjang 153 mm, lebar 48 mm, dan tinggi 35 mm sehingga mudah dibawa dan

nyaman digunakan dalam genggam tangan. Pada sisi depan prototipe terdapat layar OLED sebagai media tampilan hasil deteksi teks, tombol *toggle on/off* untuk mengaktifkan dan menonaktifkan perangkat, serta tombol interaksi yang digunakan dalam pengoperasian sistem. Sementara itu, sisi belakang perangkat dilengkapi dengan kamera ESP32-CAM sebagai komponen utama akuisisi citra, lampu *flash* untuk membantu pencahayaan saat proses deteksi, serta *port Type-C* yang digunakan untuk pengisian daya perangkat. Pada bagian atas prototipe terdapat *speaker* sebagai luaran suara hasil penerjemahan gestur BISINDO, sedangkan pada bagian bawah disediakan lubang tripod untuk mendukung stabilitas perangkat saat digunakan dalam proses pengujian maupun penggunaan langsung. Desain prototipe ini dibuat secara ringkas dan ergonomis agar mendukung mobilitas serta kemudahan penggunaan dalam aktivitas sehari-hari.

3.2. Perbandingan Performa Pelatihan Model

Proses pelatihan dilakukan menggunakan *dataset* gabungan primer dan sekunder yang telah melalui tahap pembersihan data dan *split* dengan rasio 80:10:10 untuk data latih, validasi, dan pengujian. Selain itu, pada data latih diterapkan teknik augmentasi. Evaluasi performa pelatihan dilakukan dengan membandingkan nilai *train loss* dan *validation loss* terhadap *epoch* untuk melihat stabilitas serta proses konvergensi model, sementara tingkat kemampuan deteksi dianalisis menggunakan nilai mAP50-95 terhadap *epoch*. Grafik hasil *loss* dan mAP50-95 dari masing-masing model ditunjukkan pada Gambar 5.



Gambar 5. Grafik Pelatihan Model

Berdasarkan grafik pelatihan model pada Gambar 5, seluruh model menunjukkan proses pelatihan yang berjalan dengan baik. Hal ini ditandai dengan penurunan nilai *train loss* dan *validation loss* serta peningkatan nilai mAP50-95 seiring bertambahnya *epoch*. Penurunan *loss* dan peningkatan mAP50-95 yang signifikan terjadi pada epoch awal hingga menengah, kemudian cenderung stabil setelah mendekati epoch ke-20, yang menunjukkan bahwa model mulai mencapai kondisi konvergen. Selain itu, selisih antara *train loss* dan *validation loss* pada seluruh model relatif kecil, sehingga mengindikasikan bahwa tidak terjadi *overfitting* yang signifikan selama proses pelatihan.

Secara umum, seluruh model mampu menghasilkan performa deteksi yang baik terhadap *dataset* gestur BISINDO dengan nilai mAP50-95 akhir berada pada kisaran 80%. Model YOLOv8m dan YOLOv11m menunjukkan peningkatan mAP50-95 yang lebih cepat pada tahap awal pelatihan, sedangkan YOLOv10m mengalami peningkatan yang lebih bertahap. Di sisi lain, YOLO26m menghasilkan nilai *train loss* dan *validation loss* yang paling rendah dibandingkan model lainnya, yang mengindikasikan kemampuan model dalam meminimalkan kesalahan prediksi selama proses pelatihan. Namun, berdasarkan grafik mAP50-95, YOLO26m masih menunjukkan performa yang relatif lebih rendah dibandingkan beberapa model lain. Kondisi ini dimungkinkan karena nilai *loss* yang lebih rendah dipengaruhi oleh optimasi pada komponen tertentu, seperti *bounding box loss* atau *classification loss*, sehingga tidak secara langsung menghasilkan peningkatan mAP yang paling tinggi. Oleh karena itu, hasil pelatihan ini menunjukkan bahwa setiap model memiliki karakteristik performa yang berbeda dalam proses pembelajaran dan kemampuan deteksi gestur BISINDO.

3.3. Pemilihan Model Terbaik Berdasarkan Evaluasi Kinerja Model

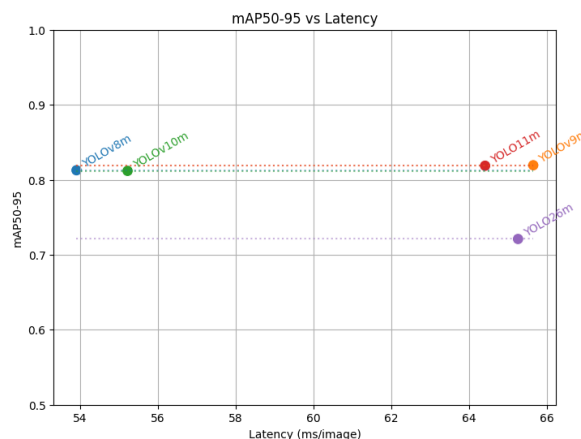
Evaluasi kinerja model dilakukan menggunakan data uji untuk mengetahui kemampuan deteksi gestur BISINDO pada kondisi yang belum pernah dilalui selama proses pelatihan. Pengujian dilakukan terhadap beberapa varian model YOLO, yaitu YOLOv8m, YOLOv9m, YOLOv10m, YOLOv11m, dan YOLO26m, dengan menggunakan sejumlah metrik performa yang mencakup *precision*, *recall*, mAP50, mAP50–95, *F1-score*, serta latensi. Penggunaan beberapa metrik tersebut bertujuan untuk memberikan penilaian yang lebih komprehensif terhadap tingkat akurasi, konsistensi deteksi, dan kecepatan inferensi model dalam mendukung implementasi sistem penerjemah BISINDO secara *real-time*. Hasil evaluasi kinerja seluruh model ditunjukkan pada Tabel 3.

Tabel 3. Metrik Evaluasi Model

Model	<i>Precision</i> (%)	<i>Recall</i> (%)	mAP50 (%)	mAP50–95 (%)	<i>F1-score</i> (%)	Latensi (ms)
YOLOv8m	97,0	96,2	98,1	81,3	96,5	53,91
YOLOv9m	97,2	96,8	98,4	81,9	96,9	65,65
YOLOv10m	95,1	95,6	98,0	81,2	95,2	55,22
YOLO11m	96,0	96,9	98,4	81,9	96,4	64,41
YOLO26m	89,6	85,6	93,9	72,1	86,2	65,26

Model YOLOv9m menunjukkan performa deteksi yang kompetitif dengan nilai *precision* sebesar 97,2%, *recall* 96,8%, mAP50 98,4%, mAP50–95 81,9%, serta *F1-score* 96,9%. Model YOLO11m juga menghasilkan performa serupa dengan nilai mAP50 sebesar 98,4% dan *recall* 96,9%. Sementara itu, YOLOv8m dan YOLOv10m menunjukkan performa deteksi yang stabil dengan nilai mAP50–95 di atas 81% serta latensi yang relatif lebih rendah dibandingkan YOLOv9m dan YOLO11m. Di antara kelima model, YOLOv8m memiliki latensi terendah sebesar 53,91 ms dengan *precision* 97,0%, *recall* 96,2%, mAP50 98,1%, mAP50–95 81,3%, dan *F1-score* 96,5%. Hasil ini menunjukkan adanya perbedaan karakteristik antara kemampuan deteksi dan kecepatan inferensi pada setiap model.

Di sisi lain, YOLO26m menunjukkan performa paling rendah pada hampir seluruh metrik evaluasi, khususnya pada mAP50–95 sebesar 72,1% dan *F1-score* 86,2%. Meskipun model ini sebelumnya menghasilkan nilai *loss* pelatihan yang rendah, hasil evaluasi menunjukkan bahwa kemampuan generalisasi dan deteksinya masih berada di bawah model lainnya. Secara keseluruhan, hasil evaluasi memperlihatkan bahwa setiap model memiliki karakteristik performa yang berbeda antara tingkat akurasi dan kecepatan inferensi. Model dengan nilai akurasi tinggi cenderung memiliki latensi yang lebih besar, sedangkan model dengan latensi lebih rendah menunjukkan performa deteksi yang sedikit lebih rendah. Oleh karena itu, analisis hubungan antara nilai mAP50–95 dan latensi diperlukan untuk menentukan keseimbangan performa yang paling sesuai pada implementasi sistem penerjemah BISINDO berbasis *real-time*. Visualisasi perbandingan tersebut ditunjukkan pada Gambar 6.



Gambar 6. Grafik Perbandingan mAP50–95 terhadap latensi Model

Berdasarkan grafik perbandingan mAP50–95 terhadap latensi pada Gambar 6, terlihat bahwa setiap model memiliki karakteristik performa yang berbeda dalam hal akurasi dan kecepatan inferensi. Model YOLOv9m dan YOLO11m menghasilkan nilai mAP50–95 tertinggi yang berada di kisaran 82%, namun keduanya memiliki latensi yang relatif tinggi, yaitu di atas 64 *ms/image*. Sementara itu, YOLO26m menunjukkan performa terendah dengan nilai mAP50–95 sekitar 72% meskipun memiliki latensi yang hampir setara dengan YOLOv9m dan YOLO11m. Di sisi lain, YOLOv10m mampu memberikan latensi yang lebih rendah, tetapi masih memiliki nilai mAP50–95 sedikit di bawah YOLOv8m.

Berdasarkan keseimbangan antara kemampuan deteksi dan efisiensi inferensi, model YOLOv8m dipilih sebagai model terbaik pada penelitian ini. Model tersebut mampu menghasilkan nilai mAP50–95 yang kompetitif sebesar 81,3% dengan latensi paling rendah, yaitu sekitar 53,91 *ms/image*. Hasil ini menunjukkan bahwa YOLOv8m memiliki performa yang lebih optimal untuk diterapkan pada sistem penerjemah BISINDO berbasis IoT yang membutuhkan respons cepat dan pemrosesan *real-time* pada perangkat dengan keterbatasan sumber daya.

3.4. Evaluasi Hasil Pengujian Deteksi Gestur BISINDO

Evaluasi pengujian dilakukan untuk mengetahui kemampuan model YOLOv8m yang telah diimplementasikan pada sistem prototipe dalam mendeteksi gestur Bahasa Isyarat Indonesia (BISINDO) secara *real-time*. Pengujian menggunakan data rekaman gestur dengan peraga yang tidak pernah muncul pada *dataset* pelatihan maupun validasi dan uji, sehingga dapat menggambarkan kemampuan generalisasi model terhadap kondisi baru. Setiap rekaman diproses melalui sistem prototipe berbasis IoT untuk mendeteksi dan mengklasifikasikan gestur BISINDO menjadi keluaran teks dan suara. Hasil pengujian ini digunakan untuk mengevaluasi konsistensi deteksi model pada penggunaan nyata, khususnya dalam kondisi variasi gerakan, posisi tangan, dan karakteristik pengguna yang berbeda. Visualisasi hasil pengujian klasifikasi gestur BISINDO pada sistem prototipe ditunjukkan pada Gambar 7.



Gambar 7. Visualisasi Hasil Pengujian Klasifikasi Gestur BISINDO

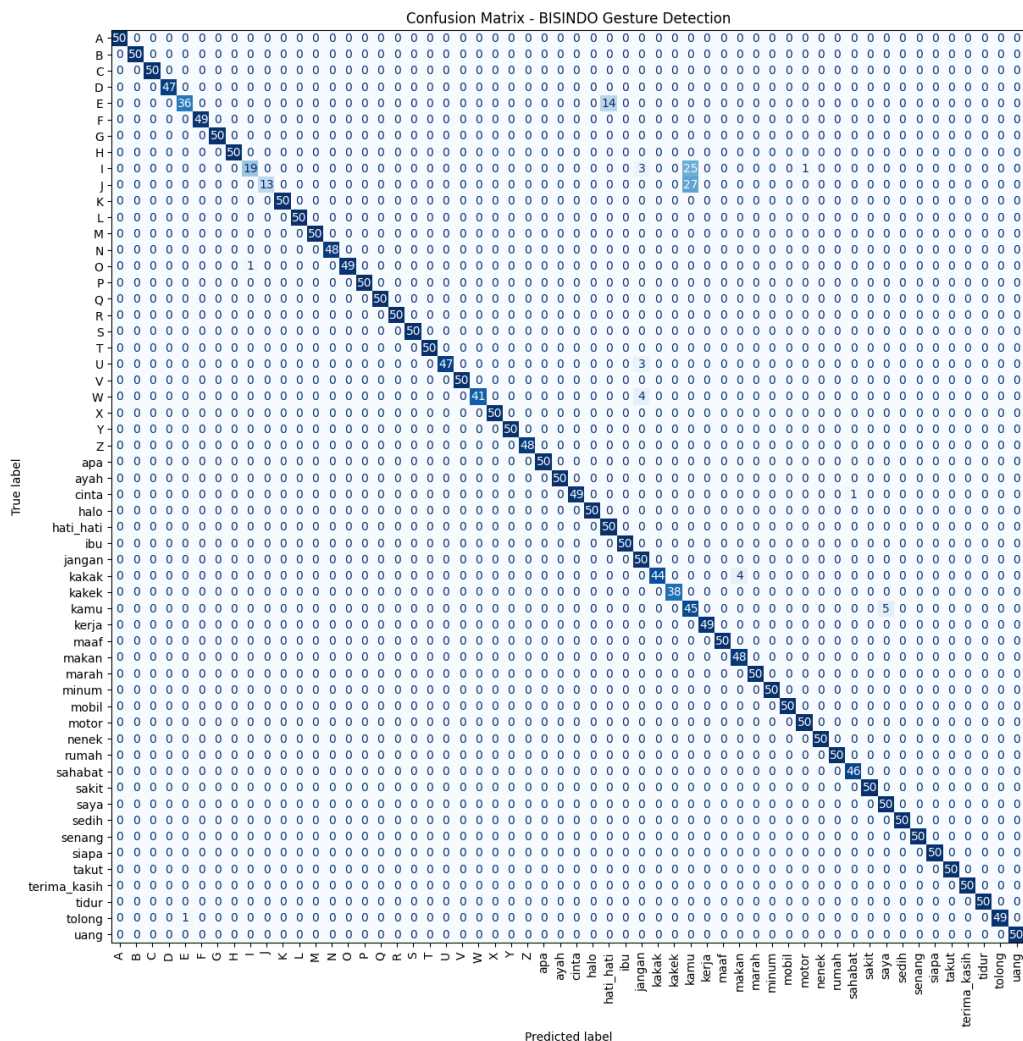
Berdasarkan visualisasi hasil pengujian pada Gambar 7, model YOLOv8m mampu mendeteksi dan mengklasifikasikan sebagian besar gestur BISINDO dengan baik pada kondisi penggunaan nyata. Beberapa gestur seperti “C” dan “ayah” berhasil dikenali dengan tingkat kepercayaan yang tinggi, yang menunjukkan bahwa model memiliki kemampuan generalisasi yang baik. Namun, pada salah satu pengujian gestur “J”, sistem belum mampu melakukan deteksi sehingga menghasilkan kondisi *no detection*. Hal ini menunjukkan bahwa masih terdapat tantangan dalam mengenali gestur tertentu yang memiliki karakteristik gerakan atau posisi tangan yang lebih kompleks. Hasil metrik evaluasi pengujian secara lengkap ditunjukkan pada Tabel 4.

Tabel 4. Metrik Evaluasi Pengujian Sistem dalam Klasifikasi Gestur BISINDO

Metrik	Nilai (%)
Accuracy	96,8
Precision	97,8
Recall	96,8
F1-score	96,6

Berdasarkan hasil evaluasi pada Tabel 4, sistem deteksi gestur BISINDO menggunakan model YOLOv8m menunjukkan performa yang baik dengan nilai akurasi sebesar 96,8%, yang menandakan bahwa sebagian besar gestur berhasil diklasifikasikan dengan benar selama pengujian. Nilai *precision* sebesar 97,8% menunjukkan bahwa prediksi yang dihasilkan model memiliki tingkat ketepatan yang

tinggi dan minim kesalahan klasifikasi. Selain itu, nilai *recall* sebesar 96,8% mengindikasikan bahwa model mampu mengenali sebagian besar gestur yang diuji, sedangkan nilai *F1-score* sebesar 96,6% menunjukkan keseimbangan yang baik antara *precision* dan *recall*. Hasil tersebut memperlihatkan bahwa model YOLOv8m memiliki kemampuan deteksi dan klasifikasi yang stabil pada implementasi sistem prototipe berbasis IoT. Visualisasi detail hasil klasifikasi dan kesalahan prediksi antar kelas ditunjukkan pada Gambar 8 melalui *confusion matrix* hasil pengujian.

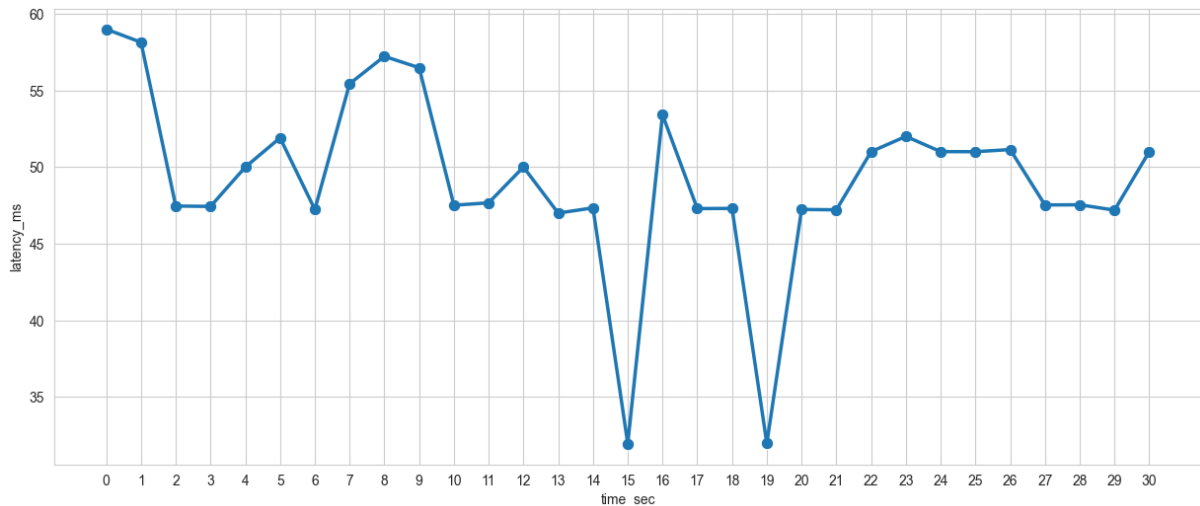


Gambar 8. Hasil *Confusion Matrix* Pengujian Sistem

Berdasarkan *confusion matrix* pada Gambar 8, model YOLOv8m menunjukkan kemampuan klasifikasi yang baik dalam mendeteksi gestur BISINDO, ditandai nilai pada diagonal utama matriks yang menunjukkan sebagian besar kelas berhasil diprediksi dengan benar. Beberapa gestur huruf dan kata dasar bahkan memperoleh hasil klasifikasi sempurna tanpa kesalahan prediksi. Namun, masih terdapat kesalahan klasifikasi pada gestur tertentu seperti “I”, “J”, “W”, “kakak”, dan “kamu”, serta beberapa kondisi *no detection*. Kesalahan tersebut disebabkan oleh kemiripan bentuk tangan antar gestur, variasi orientasi tangan, perubahan pencahayaan dan latar belakang, serta keterbatasan kualitas citra dari kamera ESP32-CAM yang memengaruhi proses ekstraksi fitur model. Meskipun demikian, secara keseluruhan *confusion matrix* menunjukkan bahwa model memiliki kemampuan generalisasi dan stabilitas deteksi yang baik pada implementasi sistem penerjemah BISINDO secara *real-time*.

3.5. Hasil Pengujian Latensi Komunikasi ESP32-CAM dengan Server

Pengujian dilakukan untuk mengevaluasi latensi komunikasi antara ESP32-CAM dan server deteksi model YOLO dalam proses penerjemahan gestur BISINDO secara *real-time*. Pengujian berlangsung selama 30 detik dengan melakukan perekaman waktu respons komunikasi dan pemrosesan data dalam satuan milidetik (ms) pada setiap interval pengujian. Nilai latensi yang diperoleh digunakan untuk mengetahui tingkat responsivitas sistem dalam mengirim citra dari ESP32-CAM menuju server serta menerima hasil deteksi kembali untuk ditampilkan sebagai keluaran sistem. Hasil pengukuran latensi selama proses komunikasi berlangsung divisualisasikan pada Gambar 9 dalam bentuk grafik latensi komunikasi ESP32-CAM dengan server deteksi model YOLO.



Gambar 9. Hasil Pengujian Latensi Komunikasi secara *Real-time*

Berdasarkan grafik latensi pada Gambar 9, komunikasi antara ESP32-CAM dan server deteksi model YOLO menunjukkan performa yang relatif stabil selama pengujian, dengan latensi rata-rata sebesar $49,22 \pm 5,74$ ms (mean $\pm$ standar deviasi). Sebagian besar nilai latensi berada pada rentang sekitar 47–52 ms, yang menunjukkan bahwa pengiriman citra dan penerimaan hasil deteksi berlangsung dengan respons yang cukup cepat untuk kebutuhan sistem *real-time*. Meskipun terdapat fluktuasi, berupa penurunan latensi hingga sekitar 32 ms dan kenaikan mendekati 59 ms pada awal pengujian, variasi tersebut tercermin dalam nilai standar deviasi sebesar 5,74 ms. Hasil ini menunjukkan bahwa komunikasi antara ESP32-CAM dan server YOLO memiliki waktu respons yang relatif konsisten dalam mendukung penerjemahan gestur BISINDO berbasis IoT secara *real-time*.

4. DISKUSI

4.1. Analisis Perbandingan Kinerja Model YOLO

Pembahasan ini berfokus pada perbandingan kinerja YOLOv8m, YOLOv9m, YOLOv10m, YOLO11m, dan YOLO26m berdasarkan metrik *precision*, *recall*, mAP50, mAP50–95, *F1-score*, dan latensi inferensi. YOLOv9m memperoleh nilai mAP50–95 tertinggi sebesar 81,9%, sedikit lebih tinggi dibandingkan YOLOv8m sebesar 81,3%. Meskipun demikian, YOLOv8m memiliki latensi inferensi terendah, yaitu 53,91 ms, sedangkan YOLOv9m membutuhkan 65,65 ms. Sebagai hasilnya, latensi YOLOv9m lebih tinggi 11,74 ms atau sekitar 21,8% dibandingkan YOLOv8m.

Perbedaan tersebut menunjukkan adanya pertukaran antara ketelitian deteksi dan kecepatan inferensi. YOLOv9m memiliki keunggulan pada mAP50–95, tetapi YOLOv8m menawarkan waktu inferensi yang lebih rendah. Dalam konteks sistem penerjemah gestur BISINDO secara *real-time*, kecepatan pemrosesan menjadi pertimbangan penting agar sistem dapat memberikan respons secara efisien. Oleh karena itu, pemilihan model perlu mempertimbangkan keseimbangan antara kinerja deteksi dan latensi, bukan hanya berdasarkan satu metrik evaluasi.

4.2. Evaluasi Implementasi YOLOv8m pada Prototipe

Berdasarkan pengujian prototipe, YOLOv8m memperoleh akurasi sebesar 96,8%, *precision* 97,8%, *recall* 96,8%, dan F1-score 96,6%. Hasil ini menunjukkan bahwa model dapat diterapkan pada sistem penerjemah gestur BISINDO yang mengintegrasikan ESP32-CAM dengan server lokal untuk menjalankan proses inferensi. Hasil deteksi selanjutnya digunakan untuk menghasilkan keluaran berupa teks dan suara.

Pada komunikasi antara ESP32-CAM dan server, diperoleh rata-rata latensi sebesar $49,22 \pm 5,74$ ms, dengan sebagian besar pengukuran berada pada rentang 47–52 ms dan fluktuasi sekitar 32–59 ms. Hasil ini menggambarkan waktu komunikasi pada sistem yang diuji. Nilai tersebut perlu dibedakan dari latensi inferensi YOLOv8m sebesar 53,91 ms karena keduanya mengukur tahapan yang berbeda. Oleh sebab itu, evaluasi waktu respons keseluruhan sebaiknya dilakukan melalui pengukuran langsung dari pengambilan citra hingga keluaran penerjemahan dihasilkan.

4.3. Analisis Kesalahan Deteksi dan Keterbatasan Sistem

Kesalahan deteksi pada pengujian ditemukan pada gestur kelas “I”, “J”, “W”, “kakak”, dan “kamu”, serta pada kondisi ketika model tidak menghasilkan deteksi. Hal ini menunjukkan bahwa meskipun model memperoleh metrik evaluasi yang tinggi, kinerja sistem masih menghadapi kendala dalam mengenali beberapa karakteristik gestur tangan.

Kesalahan tersebut disebabkan oleh beberapa faktor, yaitu kemiripan bentuk gestur antarkelas, perbedaan posisi tangan, serta rendahnya resolusi citra yang dihasilkan ESP32-CAM sehingga gambar tampak pecah dan detail jari sulit dikenali. Selain itu, gestur yang memerlukan gerakan tangan, seperti huruf “J”, lebih sulit dikenali karena bentuk gesturnya tidak hanya ditentukan oleh posisi tangan dalam satu citra, tetapi juga oleh pergerakannya. Kondisi tersebut dapat menyebabkan model salah mengklasifikasikan gestur atau tidak menghasilkan deteksi.

Temuan ini menunjukkan bahwa sistem masih memiliki keterbatasan dalam mengenali gestur dengan bentuk yang serupa dan gestur yang melibatkan gerakan. Perbaikan dapat dilakukan dengan meningkatkan kualitas dan resolusi citra masukan, memperkaya variasi dataset untuk kelas yang sering mengalami kesalahan deteksi, serta menyesuaikan metode pengenalan dengan karakteristik gestur yang diuji. Khusus untuk gestur dinamis seperti huruf “J”, penelitian selanjutnya dapat menerapkan pendekatan berbasis spasial-temporal yang menganalisis rangkaian citra dan perubahan posisi tangan dari waktu ke waktu, sehingga informasi gerakan dapat diperhitungkan dalam proses pengenalan.

4.4. Perbandingan dengan Penelitian Terdahulu

Penelitian terdahulu [8] menggunakan MobileNet V2 FPNLite SSD pada 26 kelas alfabet dari 4.054 citra, dengan *precision* 71,2%, *recall* 74,7%, dan kecepatan 30 FPS. Penelitian [9] menerapkan YOLOv8 dengan *precision* 95,8%, *recall* 97,4%, dan mAP50 99,5%. Sementara itu, penelitian [10] menggunakan sarung tangan dengan sensor fleksibel dan memperoleh tingkat keberhasilan 86,67% pada alfabet A–G, sedangkan penelitian [11] menerapkan Raspberry Pi dan BlazePose tanpa metrik kuantitatif yang memadai untuk perbandingan langsung. Penelitian [12] menggunakan CNN-LSTM untuk sepuluh kelas huruf A–J dan menghasilkan akurasi 91,4%, *precision* 89,7%, *recall* 90,5%, serta latensi komunikasi 3,1 detik. Perbedaan metode, jumlah kelas, *dataset*, dan perangkat membatasi perbandingan langsung antar penelitian tersebut.

Penelitian ini mengevaluasi lima varian YOLO pada 7.169 citra dengan 56 kelas, terdiri atas 26 huruf alfabet dan 30 kata dasar BISINDO. YOLOv8m memperoleh akurasi prototipe 96,8%, *precision* 97,8%, *recall* 96,8%, dan F1-score 96,6%, dengan mAP50 sebesar 98,1%, mAP50–95 sebesar 81,3%, serta latensi inferensi 53,91 ms. Meskipun beberapa metrik lebih tinggi dibandingkan hasil yang dilaporkan pada penelitian [8], [9], dan [12], perbedaan konfigurasi pengujian perlu diperhitungkan. Secara khusus, mAP50–90 pada [9] tidak setara dengan mAP50–95 pada penelitian ini, sedangkan FPS pada [8] dan latensi komunikasi pada [12] berbeda dari latensi inferensi per citra. Kontribusi penelitian ini terletak pada perbandingan lima varian YOLO untuk 56 kelas BISINDO dan implementasinya menggunakan ESP32-CAM yang terhubung ke server lokal, dengan YOLOv8m dipilih karena memiliki latensi inferensi terendah di antara model yang diuji.

5. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem penerjemah Bahasa Isyarat Indonesia (BISINDO) berbasis *Internet of Things* (IoT) menggunakan ESP32-CAM dan metode YOLO, dengan keluaran berupa teks pada layar OLED dan suara melalui *speaker*. Berdasarkan hasil perbandingan lima varian YOLO, YOLOv8m dipilih karena memiliki latensi inferensi terendah, yaitu 53,91 ms/citra, atau 11,74 ms lebih rendah dibandingkan YOLOv9m yang mencapai 65,65 ms/citra. Meskipun YOLOv9m menghasilkan mAP50–95 dan *F1-score* yang lebih tinggi, masing-masing sebesar 81,9% dan 96,9%, YOLOv8m tetap menunjukkan kinerja deteksi yang kompetitif dengan mAP50–95 sebesar 81,3% dan *F1-score* 96,5%. Pada pengujian prototipe, YOLOv8m memperoleh akurasi 96,8%, *precision* 97,8%, *recall* 96,8%, dan *F1-score* 96,6%. Hasil ini menunjukkan bahwa YOLOv8m sesuai untuk implementasi sistem dengan mempertimbangkan efisiensi waktu inferensi dan kinerja deteksi.

Namun, sistem masih bergantung pada server lokal untuk menjalankan inferensi sehingga kinerja penerjemahan dipengaruhi oleh komunikasi antara ESP32-CAM dan server. Kesalahan deteksi juga ditemukan pada gestur “I”, “J”, “W”, “kakak”, dan “kamu”, serta pada kondisi ketika model tidak menghasilkan deteksi. Kendala tersebut berkaitan dengan kemiripan bentuk gestur, perbedaan posisi tangan, rendahnya resolusi citra yang menyebabkan gambar pecah, dan kebutuhan untuk mengenali gerakan pada gestur dinamis seperti huruf “J”. Oleh karena itu, penelitian selanjutnya perlu memperluas *dataset* dengan menambah variasi subjek, pencahayaan, orientasi tangan, kondisi pengambilan citra, mengembangkan pengenalan gestur dinamis berbasis spasial-temporal, serta mengoptimalkan model untuk *deployment* langsung pada perangkat *edge* guna mengurangi ketergantungan pada server lokal.

DAFTAR PUSTAKA

- [1] U. Nabila and N. B. Negoro, “Pengaruh Terapi Mental Spiritual Terhadap Kesadaran Beragama Penerima Manfaat di Balai Rehabilitasi Sosial Penyandang Disabilitas Sensorik Rungu Wicara (BRSPDSRW) Melati Bambu Apus Jakarta Timur,” *JPA*, vol. 8, no. 2, pp. 125–136, Jan. 2022, doi: 10.15408/jpa.v8i2.24387.
- [2] D. Prameswari, Larasati, Muhammad Naswan Izzudin Akmal, Prismahardi Aji Riyantoko, and Dwi Arman Prasetya, “Application of Convolutional Neural Network (CNN) for Web-Based Translation of Indonesian Text into Sign Language,” *J.Apk.Impl.Sada*, vol. 1, no. 2, pp. 50–58, Oct. 2025, doi: 10.33005/jasid.v1i2.12.
- [3] D. M. Sari, A. R. Pratama, D. Pramadihanto, and B. S. Marta, “3D Object Detection Based on Point Cloud Data,” *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 7, no. 1, pp. 59–66, Jun. 2022, doi: 10.25139/inform.v7i1.4570.
- [4] A. R. Pratama, Muhammad Nasrudin, A. F. Adziima, and S. S. May Wara, “Optimalisasi Deteksi Wajah Real-Time Menggunakan HAAR Cascade Classifier berbasis OpenCV,” *JASIEK*, vol. 7, no. 1, pp. 50–61, Jun. 2025, doi: 10.26905/jasiek.v7i1.15485.
- [5] I. A. Putri, D. A. Prasetya, and T. M. Fahrudin, “Image Classification of Vine Leaf Diseases using Complex-valued Neural Network,” *jiko*, vol. 7, no. 1, pp. 36–42, Apr. 2024, doi: 10.33387/jiko.v7i1.7809.
- [6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.
- [7] A. Kamble, A. Gore, A. Kadam, and P. Gadade, “A Comprehensive Review on YOLO Object Detection Models: Evolution from YOLOv1 to YOLOv8,” *JAAFR*, vol. 3, no. 11, Nov. 2025, doi: 10.56975/jafr.v3i11.500970.
- [8] L. N. Hayati, A. N. Handayani, W. S. G. Irianto, R. A. Asmara, D. Indra, and M. Fahmi, “Classifying BISINDO Alphabet using TensorFlow Object Detection API,” *Ilk. J. Ilm.*, vol. 15, no. 2, pp. 358–364, Aug. 2023, doi: 10.33096/ilkom.v15i2.1692.358-364.
- [9] A. Bayu Pangestu, M. Rafi Muttaqin, and M. Agus Sunandar, “Sistem Deteksi Bahasa Isyarat Indonesia (BISINDO) menggunakan Algoritma You Only Look Once (YOLO)v8,” *jati*, vol. 8, no. 5, pp. 9891–9897, Sep. 2024, doi: 10.36040/jati.v8i5.10833.

- [10] N. Khamdi and M. Raja Adrafi, "Sarung Tangan Cerdas Sebagai Translator Bahasa Isyarat untuk Tuna Wicara," *elementer*, vol. 8, no. 2, pp. 113–122, Nov. 2022, doi: 10.35143/elementer.v8i2.5485.
- [11] R. Mulya Sanjaya and Y. Saragih, "Efektifitas Raspberry PI pada Alat Penerjemah Bahasa Isyarat," *JTI*, vol. 6, no. 1, pp. 130–135, Feb. 2024, doi: 10.30604/jti.v6i1.223.
- [12] I Gusti Agung Made Yoga Mahaputra, Putri Alit Widyastuti Santiary, and I Ketut Swardika, "Rancang Bangun Penerjemah BISINDO Real-time Berbasis Kamera dan Deep Learning dengan Kendali Suara ESP32 WiFi," *elementer*, vol. 11, no. 1, pp. 33–42, May 2025, doi: 10.35143/elementer.v11i1.6578.
- [13] B. A. Kumar, "Robot Vehicle with ESP32 CAM," *IJRASET*, vol. 12, no. 8, pp. 1028–1031, Aug. 2024, doi: 10.22214/ijraset.2024.64054.
- [14] P. A. Topan, D. Fardila, S. A. Rohman, S. Bahri, J. Jenal, and Y. Febriansyah, "Pemanfaatan Teknologi Arduino dan DFPlayer Mini untuk Perangkat Pemutar Audio di Masjid Raudhatul Jannah Desa Gontar, Kabupaten Sumbawa, Nusa Tenggara Barat," *JAI*, vol. 9, no. 4, pp. 1797–1807, Dec. 2022, doi: 10.29303/abdiinsani.v9i4.829.
- [15] K. V. S. S. Ganesh, S. P. S. Jeyanth, and A. R. Bevi, "IOT based portable heart rate and SpO2 pulse oximeter," *HardwareX*, vol. 11, Apr. 2022, doi: 10.1016/j.ohx.2022.e00309.
- [16] bisindoskripsi, "Dataset BISINDO 1," Roboflow. Accessed: May 06, 2026. [Online]. Available: <https://universe.roboflow.com/bisindoskripsi/bisindo-skripsi-fix>
- [17] yolotga, "Dataset BISINDO 2," Roboflow. Accessed: May 06, 2026. [Online]. Available: <https://universe.roboflow.com/yolotga-mevpn/tga-bisindo-uf0tu>
- [18] S. Isnaniah, T. Agustina, Islahuddin, and F. Annisa, "The Use of Sign Language in Deaf Indonesian Classrooms in Surakarta," *KEMBARA*, vol. 9, no. 2, pp. 468–481, Oct. 2023, doi: 10.22219/kembara.v9i2.25990.
- [19] E. Price and A. Ahmad, "Accelerated Video Annotation Driven by Deep Detector and Tracker," in *Intelligent Autonomous Systems 18*, vol. 794, S.-G. Lee, J. An, N. Y. Chong, M. Strand, and J. H. Kim, Eds., in *Lecture Notes in Networks and Systems*, vol. 794, Cham: Springer Nature Switzerland, 2024, pp. 141–153. doi: 10.1007/978-3-031-44981-9_12.
- [20] G. Jocher, J. Qiu, M. Liu, S. Lyu, F. C. Akyon, and M. E. Kalfaoglu, "Ultralytics YOLO26: Unified Real-Time End-to-End Vision Models." 2026. doi: 10.48550/arXiv.2606.03748.
- [21] G. Jocher, J. Qiu, and A. Chaurasia, *Ultralytics YOLO*. (Jan. 2023). [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [22] H. Prasetya, R. R. Balo, T. Tumbal, A. M. Sambul, and M. Dwisnanto Putro, "Comparative Study of Lightweight YOLOv12 Models for Real-Time Underwater Object Detection," *J. Telematika*, vol. 20, no. 2, pp. 127–137, Jan. 2026, doi: 10.61769/telematika.v20i2.799.
- [23] D. Niranjana, B. VinayKarthik, and Mohana, "Performance Analysis of SSD and Faster RCNN Multi-class Object Detection Model for Autonomous Driving Vehicle Research Using CARLA Simulator," in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Sep. 2021, pp. 1–6. doi: 10.1109/ICECCT52121.2021.9616712.
- [24] S. Utomo, S. Sigit, N. Sulistyowati, D. A. Prasetya, and T. M. Fahrudin, "Perbandingan Kinerja Algoritma XGBoost dan CatBoost dalam Klasifikasi Risiko Penyakit Diabetes," *alinier*, vol. 6, no. 2, pp. 83–96, Nov. 2025, doi: 10.36040/aliner.v6i2.14772.