

DETEKSI ABJAD BAHASA ISYARAT BISINDO *REAL-TIME* PADA SMARTPHONE MENGGUNAKAN YOLOV11

Rivaldo Nugraha ¹⁾, Rofik Adam Nugraha ²⁾, Muhammad Abdul Mujib ³⁾,
Okyza Maherdy Prabowo ⁴⁾, Anggi Cipta Lestari ⁵⁾, Shafira Febriani ⁶⁾

^{1,2,3,6)} Teknik Informatika STMIK Amik Bandung

^{4,5)} Sistem Informasi STMIK Amik Bandung

email : Rivaldo@stmik-amikbandung.ac.id ¹⁾, Rofikadamnugraha@gmail.com ²⁾, Abdul.mujib@stmik-amikbandung.ac.id ³⁾,
Okyza@stmik-amikbandung.ac.id ⁴⁾, Anggi@stmik-amikbandung.ac.id ⁵⁾, Shafira@stmik-amikbandung.ac.id ⁶⁾

Abstraksi

Bahasa isyarat merupakan sarana komunikasi utama bagi komunitas tunarungu dan penyandang gangguan pendengaran, sekaligus berperan penting dalam meningkatkan inklusivitas sosial dan menjembatani kesenjangan komunikasi dengan masyarakat umum. Seiring meningkatnya jumlah penyandang gangguan pendengaran, diperlukan solusi komunikasi yang mudah diakses dan bersifat adaptif. Perkembangan kecerdasan buatan dan pembelajaran mesin memberikan peluang untuk mengotomatisasi deteksi bahasa isyarat secara *real-time*. Penelitian ini mengembangkan sistem deteksi abjad BISINDO A–Z menggunakan dua pendekatan, yaitu YOLOv11 dan RF-DETR. YOLOv11 dipilih karena arsitekturnya ringan dan efisien untuk perangkat mobile, sedangkan RF-DETR digunakan sebagai pembandingan berbasis Transformer dengan tingkat akurasi tinggi. Hasil eksperimen menunjukkan bahwa RF-DETR mencapai akurasi lebih tinggi dengan mAP@50 sebesar 99,8%, namun terbatas pada implementasi berbasis web. YOLOv11 memperoleh mAP@50 sebesar 99,4% dan berhasil diimplementasikan pada smartphone Android dengan kinerja *real-time* yang responsif. Temuan ini menunjukkan bahwa YOLOv11 lebih layak untuk aplikasi deteksi abjad BISINDO berbasis perangkat mobile.

Kata Kunci :

BISINDO, abjad bahasa isyarat, YOLOv11, RF-DETR, deteksi real-time.

Abstract

Sign language plays a crucial role as the primary communication medium for the deaf and hard-of-hearing community, while also promoting social inclusion and bridging communication gaps with the general public. As the number of individuals with hearing impairments is projected to increase, accessible and adaptive communication solutions are increasingly needed. Advances in artificial intelligence and machine learning provide opportunities to automate sign language detection in real-time. This study focuses on developing a real-time detection system for BISINDO (Indonesian Sign Language) alphabet A–Z using two approaches: YOLOv11 and RF-DETR. YOLOv11 is selected for its lightweight architecture and efficiency on mobile devices, while RF-DETR serves as a comparative model based on Transformer architecture with high detection accuracy. Experimental results show that RF-DETR achieves slightly higher performance with an mAP@50 of 99.8%, but is limited to web-based implementation. In contrast, YOLOv11 achieves an mAP@50 of 99.4% and is successfully deployed on Android smartphones with responsive real-time performance. These findings indicate that YOLOv11 is more feasible for mobile-based BISINDO alphabet detection applications.

Keywords :

BISINDO, sign language alphabet, YOLOv11, RF-DETR, real-time detection.

Pendahuluan

Komunikasi merupakan aspek fundamental dalam kehidupan manusia yang memungkinkan pertukaran informasi, ide, dan emosi. Bagi komunitas tunarungu, bahasa isyarat adalah sarana komunikasi utama; namun tidak semua orang dapat memahami atau mengaksesnya secara luas. Oleh sebab itu, diperlukan solusi teknologi yang mampu meningkatkan aksesibilitas komunikasi, salah satunya melalui sistem pengenalan gestur abjad bahasa isyarat yang efisien dan *portabel*. Deteksi gestur bahasa isyarat dapat meningkatkan aksesibilitas komunikasi bagi penyandang tunarungu dengan mengotomatisasi translasi gestur menjadi

label atau teks, dan topik ini telah menjadi fokus banyak penelitian karena potensi dampaknya sosialnya [1].

Kemajuan dalam kecerdasan buatan dan pembelajaran mesin membuka peluang untuk mengotomatisasi pengenalan gestur. Metode-metode berbasis *deep learning*, khususnya detektor objek yang berbasis CNN seperti keluarga YOLO, banyak dipakai untuk tugas pengenalan huruf/gestur isyarat karena kemampuan *real-time* dan efisiensi komputasi yang baik pada perangkat *edge* [2]. Di sisi lain, arsitektur berbasis Transformer seperti RF-DETR menunjukkan keunggulan dalam memahami

konteks visual kompleks dan sering kali mencapai akurasi yang lebih tinggi pada tugas-tugas deteksi. Selain itu, ketersediaan *dataset* khusus untuk abjad BISINDO (A–Z) memungkinkan pelatihan *model* yang disesuaikan dengan variasi lokal (latar, perangkat, subjek), sehingga penelitian yang memfokuskan pada deteksi abjad BISINDO praktis untuk dilakukan dan direproduksi [3]. Berdasarkan tinjauan tersebut, terdapat peluang untuk mengembangkan sistem yang fokus pada deteksi abjad BISINDO (A–Z) secara *real-time* pada perangkat *smartphone*. Pendekatan ini menuntut keseimbangan antara akurasi *model* dan kelayakan implementasi di perangkat *mobile* (*latency*, ukuran *model*, dan konsumsi sumber daya). Oleh karena itu, penelitian ini menggunakan YOLOv11 sebagai *model* utama karena arsitekturnya yang ringan dan cocok untuk *deploy* di perangkat *mobile*, serta menggunakan RF-DETR sebagai pembanding untuk mengevaluasi perbedaan akurasi dan trade-off implementasi antara kedua arsitektur. Dengan demikian, penelitian ini bertujuan mengembangkan dan menguji model deteksi abjad BISINDO (A–Z) secara *real-time* pada *smartphone* menggunakan YOLOv11 serta menganalisis perbandingan kinerjanya dengan RF-DETR, baik dari sisi metrik deteksi maupun aspek implementasi (*mobile* vs *web*).

Tinjauan Pustaka

Berbagai penelitian terdahulu terkait pengenalan bahasa isyarat, penggunaan algoritma YOLO, dan penerapan *deep learning* untuk deteksi objek menjadi dasar penting dalam menilai dan membandingkan metode serta hasil penelitian ini. Virgiyawan (2024) [4] mengembangkan sistem pengenalan bahasa isyarat berbasis deteksi objek menggunakan YOLOv8 untuk mengidentifikasi abjad Sistem Isyarat Bahasa Indonesia (SIBI), dengan kemampuan deteksi *real-time* dan tingkat kepercayaan yang cukup tinggi, seperti pada huruf “A” yang mencapai 87%. Maulida (2023) [5] mengevaluasi akurasi deteksi simbol abjad SIBI dengan pendekatan CNN menggunakan model pra-latih SSD MobileNet dan YOLOv7, serta menemukan bahwa YOLOv7 memberikan performa lebih baik dengan akurasi 100% dibandingkan SSD MobileNet sebesar 98,07%. Penelitian oleh Imam, Aeni, dan Fathulloh (2023) [6] memanfaatkan YOLOv5s untuk mendeteksi 10 gestur abjad SIBI berbasis 3.000 citra dan menghasilkan akurasi *real-time* sebesar 94,5%. Selanjutnya, Febrina Silalahi dkk. (2024) [7] mengembangkan aplikasi *mobile* penerjemah BISINDO berbasis YOLOv5 dan menunjukkan bahwa faktor pencahayaan sangat memengaruhi kinerja sistem, dengan deteksi optimal pada kondisi cahaya 100% dan penurunan signifikan pada pencahayaan rendah. Luthfy, Setianingsih, dan Paryasto (2023) [8] mengembangkan sistem klasifikasi BISINDO menggunakan YOLO dengan konfigurasi pelatihan terbaik yang menghasilkan

nilai mAP@0.5 IoU sebesar 99,27%. Selain itu, Halim dan Lina (2023) [9] mengembangkan aplikasi identifikasi bahasa isyarat berbasis gerak tubuh secara *real-time* menggunakan YOLO untuk mengenali 50 gestur BISINDO, dengan akurasi terbaik sebesar 67% pada model yang dilatih selama 200 *epoch*. Di sisi lain, Sapkota dkk. (2025) [10] membandingkan arsitektur Transformer (RF-DETR) dan CNN (YOLOv12) pada deteksi objek di lingkungan kompleks, dan menemukan bahwa RF-DETR lebih unggul dalam pemodelan konteks global serta penanganan oklusi dan kamuflase, dengan capaian mAP@50 sebesar 0,9464 untuk satu kelas dan 0,8298 untuk multi-kelas.

Deep learning

Deep learning merupakan cabang dari Machine Learning yang termasuk dalam Kecerdasan Buatan (AI), dengan memanfaatkan jaringan saraf tiruan berlapis banyak untuk memproses data kompleks seperti gambar, suara, dan teks. Pendekatan ini memungkinkan komputer belajar secara otomatis dari data tanpa pemrograman eksplisit, sehingga efektif untuk pengembangan sistem AI yang canggih. Beberapa algoritma yang umum digunakan antara lain *Recurrent Neural Network*, *Convolutional Neural Network*, *Self-Organizing Maps*, dan *Long Short-Term Memory*, yang mampu menghasilkan output secara tepat dan akurat [11].

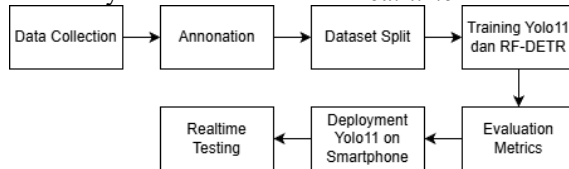
Layer

Layer merupakan unit struktural utama dalam jaringan yang berperan penting dalam pengolahan informasi. Setiap layer menerima aliran data sebagai masukan dari layer sebelumnya. Data tersebut kemudian diproses melalui serangkaian perhitungan internal. Hasil proses ini berupa representasi data baru yang telah ditransformasikan. Representasi tersebut selanjutnya diteruskan ke layer berikutnya dalam jaringan. Proses bertahap ini memungkinkan sistem menyelesaikan tugas seperti pengenalan pola, klasifikasi data, dan pembuatan prediksi [12].

Berdasarkan penelitian sebelumnya, sebagian besar studi tentang deteksi bahasa isyarat menggunakan arsitektur YOLO yang fokus pada SIBI atau BISINDO dalam skenario berbasis desktop atau web, dan tidak membahas implementasi *real-time* pada perangkat *smartphone*. Selain itu, belum ada penelitian yang secara langsung membandingkan arsitektur CNN (YOLOv11) dan Transformer (RF-DETR) pada *dataset* BISINDO A–Z dalam lingkungan pelatihan yang sama. Penelitian ini mengatasi celah tersebut dengan tiga hal utama, yaitu (1) membangun *dataset* BISINDO yang diperluas, (2) melatih YOLOv11 dan RF-DETR pada platform yang sama, serta (3) mengevaluasi aspek akurasi dan kemungkinan implementasi pada perangkat *mobile*. Dengan demikian, kontribusi utama penelitian ini terletak pada analisis keseimbangan antara akurasi dan kemudahan penggunaan model untuk aplikasi deteksi BISINDO *real-time* yang berbasis *smartphone*.

Metode Penelitian

Penelitian ini menggunakan Metodologi Penelitian Eksperimental Kuantitatif. Pendekatan ini dipilih karena tujuan utama penelitian adalah untuk mengukur dan membandingkan kinerja dari dua arsitektur *model deep learning* yang berbeda dalam menyelesaikan tugas yang sama, yaitu deteksi abjad bahasa isyarat BISINDO secara *real-time*.



Gambar 1. Alur Diagram Penelitian

Alur diagram penelitian ditunjukkan pada gambar 1, dimana menggambarkan tahapan mulai dari pengumpulan *dataset*, proses anotasi dan pembagian data, pelatihan model YOLOv11 dan RF-DETR, evaluasi performa, hingga implementasi dan pengujian sistem deteksi *real-time* pada smartphone. Dimana eksperimen ini dilakukan dengan membandingkan dua arsitektur model, yaitu YOLOv11 dan RF-DETR, sebagai variabel utama yang diuji. Kinerja masing-masing model diukur berdasarkan kemampuan deteksi, baik melalui metrik kuantitatif seperti akurasi, presisi, dan recall, maupun melalui pengujian langsung terhadap keberhasilan pengenalan gestur. Untuk menjaga keadilan dan konsistensi pengujian, kedua model menggunakan *dataset* yang sama, dilatih pada platform dan metode yang identik (Roboflow Train), serta menerapkan proses augmentasi dan pembagian data *training*, validasi, dan *testing* yang seragam, sehingga perbedaan hasil yang diperoleh dapat dikaitkan langsung dengan perbedaan arsitektur model.

Perangkat Penelitian

Penelitian ini memanfaatkan Smartphone Poco M3 sebagai perangkat keras (*hardware*) untuk implementasi dan pengujian kinerja model YOLOv11. Proses pelatihan model YOLOv11 dan RF-DETR dilakukan pada infrastruktur cloud yang disediakan oleh platform Roboflow. Platform ini secara otomatis mengalokasikan sumber daya komputasi, termasuk *High-Performance GPU*, untuk menjalankan proses pelatihan secara efisien [13]. Dengan demikian, perangkat keras untuk tahap pelatihan tidak bergantung pada spesifikasi komputer peneliti.

Alur Kerja Sistem

Perangkaian Alur kerja sistem dari Deteksi Bahasa Isyarat yaitu Dengan dirancang dengan modul berurutan dari pengambilan citra (*Capture*) hingga penyajian hasil deteksi (*output*). Secara keseluruhan alur sistem kerja sebagai berikut :

Pada tahap awal, aplikasi memanfaatkan API kamera (*Camera2*) pada perangkat *Android* untuk mengambil *frame video* secara terus-menerus. Setiap *frame* dikonversi menjadi format *Bitmap* atau

Image yang siap diproses lebih lanjut. Pengambilan *frame* dilakukan pada resolusi standar (misalnya $720 \times 1280 \text{ pixel}$) untuk menyeimbangkan kebutuhan akurasi dan kinerja.

Setiap *frame* yang diterima kemudian menjalani serangkaian langkah *preprocessing* agar sesuai dengan *input* yang diharapkan oleh model YOLOv11:

1. *Resize*: *Frame* diubah ukurannya menjadi $640 \times 640 \text{ pixel}$ menggunakan algoritma *letterbox*. Metode ini mempertahankan rasio aspek gambar asli dengan menambahkan *padding* jika diperlukan.
2. *Normalize*: Nilai piksel diubah dari rentang $[0, 255]$ menjadi $[0, 1]$ dengan membagi setiap elemen piksel dengan 255.

Frame hasil *preprocessing* dikonversi menjadi tensor berdimensi $1 \times 640 \times 640 \times 3$, lalu diproses melalui jaringan YOLOv11 yang telah dilatih sebelumnya. Tahapan ini mencakup *Anchor Box Prediction* pada setiap grid sel dan di berbagai skala, model menghasilkan sejumlah *anchor box* yang berisi informasi koordinat (x, y, w, h), *confidence score*, dan label kelas *gesture*.

Hasil keluaran mentah (*raw outputs*) dari model diproses lebih lanjut untuk menghasilkan prediksi yang bersih dan tidak tumpang tindih:

1. *Non-Max Suppression (NMS)*: Mengeliminasi kotak prediksi yang memiliki tumpang tindih (*IoU*) tinggi sehingga hanya kotak dengan *confidence score* tertinggi per objek yang dipertahankan.
2. *Confidence Thresholding*: Menyingkirkan semua prediksi dengan skor keyakinan di bawah ambang batas (misal 0,5) guna mengurangi false positives.

Kotak prediksi akhir beserta label *gesture* ditampilkan di atas video feed secara *real-time*. Implementasi output meliputi: *Overlay Bounding box & Label*: Menggambar kotak transparan berwarna di sekitar area tangan pengguna disertai penulisan label (misal "A").

Pengumpulan Dataset

Dalam tahap perencanaan *dataset*, Untuk membangun model deteksi yang kuat dan mampu melakukan generalisasi dengan baik, penelitian ini mengadopsi pendekatan strategi pengumpulan data hibrida, yang menggabungkan dua sumber utama: *dataset* publik yang berskala besar dan *dataset* mandiri yang dikumpulkan secara spesifik untuk penelitian ini.

Perencanaan Dataset

Perencanaan *dataset* dirancang untuk memastikan model dilatih dengan data yang tidak hanya banyak secara kuantitas, tetapi juga kaya akan variasi. Oleh karena itu, strategi pengumpulan data dibagi menjadi dua:

1. *Akuisisi Dataset Primer* (Sumber Publik)

Citra gambar merupakan representasi visual dari objek, pemandangan, atau informasi yang direkam

secara digital. Dalam pengolahan citra digital, citra dipahami sebagai data dua dimensi [14].

Penulis memanfaatkan Kaggle sebagai sumber utama karena menyediakan koleksi citra dalam jumlah besar dan telah terkurasi untuk gesture huruf A-Z. salah satu *dataset* utama yang digunakan adalah kumpulan BISINDO karya Agung Ma'ruf [15], yang sejatinya merupakan gabungan dari *dataset* lainnya yang tersedia di kaggle yang telah terkurasi dan mencakup berbagai variasi gestur. Total citra yang berhasil dikumpulkan dari sumber ini adalah 11.841 citra.

2. Akuisisi *Dataset* Sekunder (Sumber Mandiri)

Untuk mengatasi potensi bias pada *dataset* publik dan meningkatkan kemampuan *model* dalam mengenali gestur pada kondisi yang lebih beragam (cahaya, latar belakang) serta gestur dari tangan peneliti sendiri, dilakukan pengambilan data secara mandiri. Penulis melakukan perekaman *video* untuk setiap gestur abjad A-Z menggunakan kamera *smartphone* Poco M3. Proses perekaman sengaja dilakukan dalam beberapa sesi dengan kondisi yang berbeda-beda untuk memaksimalkan variasi, meliputi:

- Pencahayaan: Cahaya alami (dekat jendela), cahaya lampu ruangan (malam hari), dan kondisi sedikit redup.
- Latar Belakang: Dinding polos, latar belakang dengan objek (misal: rak buku), dan luar ruangan.

Video yang telah direkam kemudian diproses untuk diekstraksi menjadi gambar (*frame*). Dari setiap detik *video*, dipilih 2-3 *frame* terbaik yang tidak buram (*blur*) dan menunjukkan gestur dengan jelas untuk menghindari ketidakjelasan data.

Seluruh 963 citra baru ini kemudian diunggah ke proyek yang sama di Roboflow. Penulis melakukan proses anotasi secara manual dengan menggambar *bounding box* di sekitar area tangan dan memberikan label kelas yang sesuai (A-Z). Total akhir *dataset* ini berjumlah 936 citra yang diekstraksi dari rekaman *video*.

Kombinasi kedua sumber ini direncanakan untuk menghasilkan total 26 kelas (huruf A-Z) dengan variabel eksperimen yang lebih luas, mencakup latar belakang polos, bercorak, kondisi pencahayaan terang dan redup, serta variasi sudut pengambilan gambar.

Akuisisi Data

Setelah mengunduh kumpulan citra awal dari Kaggle, seluruh file kemudian diunggah ke *platform Roboflow* untuk mempermudah proses anotasi dan manajemen data. *Roboflow* menyediakan antarmuka yang intuitif untuk menggambar *bounding box*, mengelola label.

Lebih jauh lagi, melalui fitur *Roboflow Universe*, penulis menambahkan beberapa *dataset* eksternal tambahan yang relevan untuk meningkatkan ragam latar belakang, variasi pencahayaan, dan rentang warna kulit. Dengan demikian, kumpulan data

BISINDO yang digunakan dalam penelitian ini tidak hanya berasal dari satu sumber, tetapi telah diperkaya dengan variasi citra yang lebih luas guna memperkuat kemampuan generalisasi *model* deteksi [16].

Pembagian *Dataset*

Total sampel yang berhasil dikumpulkan mencapai sekitar 12.777 citra, dengan masing-masing kelas memiliki minimal 450++ contoh. Sebagian besar citra telah berstatus terannotasi.

Dataset kemudian dibagi menjadi subset pelatihan (75 %), validasi (13 %), dan pengujian (13 %) untuk mendukung evaluasi kinerja *model* secara menyeluruh. Subset pelatihan (*training*) digunakan untuk melatih *model*, subset validasi (*validation*) digunakan untuk memantau kinerja *model* selama pelatihan dan melakukan penyesuaian *hyperparameter*, sementara subset pengujian (*testing*) digunakan untuk evaluasi akhir *model* yang belum pernah dilihat sebelumnya.

Pelatihan Model Deteksi Menggunakan Roboflow Train

Setelah tahap persiapan dan pembagian *dataset* selesai dilakukan di dalam *platform Roboflow*, langkah selanjutnya adalah melatih *model* deteksi objek. Untuk tujuan ini, penelitian ini memanfaatkan fitur *Roboflow Train*, sebuah layanan pelatihan berbasis cloud yang dirancang untuk mengotomatisasi dan menyederhanakan proses pelatihan *model* visi komputer. Pemilihan *platform* ini didasarkan pada kemampuannya untuk mengelola keseluruhan alur kerja secara efisien, mulai dari penyiapan data hingga *deployment model*.

Proses Pelatihan

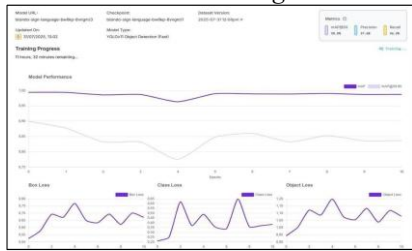
Proses pelatihan diinisiasi langsung dari antarmuka *Roboflow* setelah sebuah versi *dataset* yang telah diaugmentasi siap digunakan. *Roboflow Train* menyediakan opsi "*Custom Train*" yang memungkinkan pengguna untuk melatih *model* tanpa perlu melakukan konfigurasi manual yang rumit terkait infrastruktur komputasi dan *hyperparameter*.

Tahap konfigurasi ini meliputi:

- 1) Pemilihan Arsitektur *Model*: *Roboflow* menyediakan berbagai pilihan arsitektur *model* yang sudah dioptimalkan. Pilihan ini mencakup *model-model* canggih seperti *RF-DETR* (berbasis *Transformer*), *Roboflow 3.0* (*model* general-purpose), serta berbagai varian *YOLO*, termasuk *YOLO-NAS* dan iterasi terbaru seperti *YOLOv11*. Dalam penelitian ini, arsitektur *YOLOv11* dipilih sebagai *model* utama.
- 2) Pelatihan Otomatis (*Automated Training*): *Platform* ini secara otomatis menangani alokasi sumber daya komputasi, seperti penyediaan *cloud GPU*, dan menentukan konfigurasi pelatihan yang stabil dan efisien. Hal ini membebaskan peneliti dari tugas penyiapan lingkungan yang kompleks.

- 3) *Transfer Learning*: Secara default, pelatihan dimulai dari checkpoint *model* yang telah dilatih pada *dataset* besar (umumnya *COCO*), sebuah teknik yang dikenal sebagai *transfer learning*. Pendekatan ini secara signifikan mempercepat waktu konvergensi dan sering kali menghasilkan *model* dengan akurasi yang lebih tinggi.
- 4) *Sistem Kredit*: Untuk menjalankan proses pelatihan, *Roboflow* menggunakan sistem berbasis kredit, di mana setiap sesi pelatihan akan mengonsumsi sejumlah kredit yang tersedia di akun pengguna.

Proses Pelatihan dan Monitoring



Gambar 2. Dashboard Monitoring Pelatihan Model

Sumber: Hasil Pelatihan Roboflow (2025)

Berdasarkan gambar 2, dimana setelah konfigurasi selesai dan pelatihan dimulai, seluruh proses komputasi berjalan di *server cloud Roboflow*. Pengguna tidak perlu menjaga koneksi internet tetap stabil atau membiarkan komputer lokal menyala. Kemajuan proses pelatihan dapat dipantau secara *real-time* melalui sebuah dashboard interaktif. *Dashboard* ini menampilkan grafik-grafik penting. Terdapat Grafik penting yaitu:

- 1) Grafik mean Average Precision (mAP) per *epoch*.
- 2) Grafik Precision dan Recall.
- 3) Grafik *Loss* (kerugian) untuk *bounding box*, klasifikasi, dan objek

Visualisasi ini sangat penting untuk memahami bagaimana model belajar dari waktu ke waktu dan untuk mengidentifikasi apakah proses pelatihan berjalan dengan baik.

Metode Pengujian

Untuk memberikan tolok ukur (*benchmark*) dan mengevaluasi kinerja *model YOLOv11* secara komprehensif, penelitian ini melakukan eksperimen perbandingan dengan *model* yang menggunakan arsitektur *RF-DETR* (*Roboflow Detection Transformer*).

Pengujian Kuantitatif

Pengujian ini bertujuan untuk mendapatkan hasil yang objektif berupa angka. Data untuk pengujian ini diambil dari hasil akhir pelatihan di platform Roboflow. Metrik yang digunakan untuk menilai performa *model* adalah:

1. *mAP@50*: Angka yang menunjukkan tingkat akurasi umum dari model dalam mendeteksi gestur.

2. *Precision*: Mengukur seberapa akurat tebakan model. Dari semua deteksi yang dibuat, berapa persen yang benar.
3. *Recall*: Mengukur kelengkapan deteksi model. Dari semua gestur yang seharusnya ada, berapa persen yang berhasil ditemukan.
4. *F1-Score*: Angka yang menyeimbangkan antara *Precision* dan *Recall* untuk memberikan gambaran performa yang menyeluruh.
5. *Confusion Matrix*: Sebuah tabel yang digunakan untuk melihat secara rinci jenis kesalahan yang dibuat oleh model (misalnya, gestur apa yang sering tertukar).

Pengujian Kualitatif

Pengujian ini fokus pada bagaimana performa model saat dicoba secara praktis di dunia nyata, tidak hanya di atas kertas:

1. Proses Pengujian
 - a. Model *YOLOv11* diuji coba secara langsung pada aplikasi smartphone Android (Poco M3) untuk melihat kinerjanya secara *real-time*.
 - b. Model *RF-DETR* diuji dengan cara mengunggah gambar-gambar dari test set ke fitur "*Preview*" di Roboflow, karena adanya kendala teknis untuk dijalankan langsung di smartphone.
2. Hal yang diamati
 - a. Apakah model berhasil mendeteksi semua 26 huruf abjad?
 - b. Seberapa tinggi skor keyakinan (*confidence score*) yang muncul saat model mendeteksi gestur?

Pelatihan Model Deteksi Menggunakan Roboflow Train

Untuk memberikan tolok ukur (*benchmark*) dan mengevaluasi kinerja *model YOLOv11* secara komprehensif, penelitian ini melakukan eksperimen perbandingan dengan *model* yang menggunakan arsitektur *RF-DETR* (*Roboflow Detection Transformer*).

Deskripsi Dan Tujuan Pemilihan Model Pembandingan

Pengujian ini bertujuan untuk mendapatkan hasil yang objektif berupa angka. Data untuk pengujian ini diambil dari hasil akhir pelatihan di platform Roboflow. Metrik yang digunakan untuk menilai performa *model* adalah:

1. Perbandingan Arsitektur Modern: Membandingkan dua pendekatan canggih dalam deteksi objek: pendekatan berbasis CNN (*YOLOv11*) yang unggul dalam efisiensi dan ekstraksi fitur lokal, dengan pendekatan berbasis Transformer (*RF-DETR*) yang unggul dalam memahami konteks global sebuah gambar..
2. Keadilan Eksperimen: Karena *RF-DETR* juga merupakan model yang didukung dan dapat dilatih langsung di platform Roboflow, hal ini

memungkinkan dilakukannya perbandingan yang sangat adil. Kedua model dapat dilatih menggunakan *dataset*, augmentasi, dan lingkungan komputasi yang identik.

Kontras arsitektural ini menjadikan *RF-DETR* sebagai *baseline* yang ideal untuk mengukur keunggulan dan menganalisis *trade-off* dari *YOLOv11* dalam tugas deteksi bahasa isyarat secara spesifik.

Skenario Dan Prosedur Pengujian

Untuk memastikan perbandingan yang valid dan adil, kedua model (*YOLOv11* dan *RF-DETR*) dilatih dan diuji dalam skenario dan prosedur yang terstandarisasi:

1. Platform Pelatihan: Kedua model dilatih menggunakan layanan Roboflow Train untuk menjamin konsistensi lingkungan pelatihan..
2. *Dataset*: Keduanya menggunakan versi *dataset* yang sama persis, termasuk di dalamnya konfigurasi augmentasi dan rasio pembagian data latih, validasi, dan uji..
3. Prosedur Pengujian: Untuk *YOLOv11*, pengujian kinerja kuantitatif (pengukuran FPS) dan kualitatif (analisis visual) dilakukan secara langsung pada aplikasi *real-time* di smartphone Poco M3. Untuk *RF-DETR*, karena menghadapi tantangan teknis dalam konversi dan implementasi ke perangkat mobile, pengujian difokuskan pada evaluasi kualitatif. Pengujian ini dilakukan dengan menggunakan fitur "Preview Model" pada platform Roboflow, di mana gambar-gambar dari set data uji yang sama diunggah untuk diobservasi dan dianalisis kemampuan deteksinya secara visual."

Hasil dan Pembahasan

Hasil Pelatihan Model Deteksi

Tahap pelatihan merupakan langkah krusial untuk menghasilkan *model* yang mampu mengenali gestur bahasa isyarat BISINDO dengan akurat. Berikut adalah hasil akhir dari proses pelatihan untuk kedua *model* yang dievaluasi.

Kinerja Pelatihan Model YOLOv11

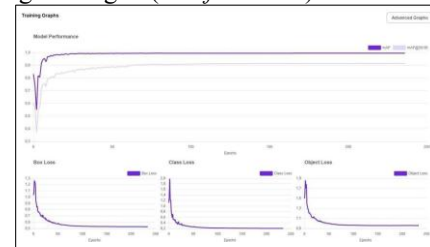


Gambar 3. Metrics hasil mAp@50, Precision, Recall

Sumber: Hasil Pelatihan Roboflow (2025)

Berdasarkan Gambar 3, *model YOLOv11* berhasil mencapai *mAP@50* sebesar 99.4%, Presisi sebesar 99.5%, dan *Recall* sebesar 99.4%. Nilai-nilai ini mengindikasikan bahwa *model* memiliki kemampuan yang nyaris sempurna dalam mendeteksi dan mengklasifikasikan gestur abjad BISINDO dengan benar pada set data yang digunakan. Gambar 2 menyajikan visualisasi

komprehensif dari dinamika proses pelatihan *model YOLOv11* yang berjalan selama kurang lebih 250 *epoch*. Grafik-grafik ini memberikan wawasan mendalam mengenai bagaimana *model* belajar dan beradaptasi dengan *dataset* bahasa isyarat BISINDO dari waktu ke waktu, yang terbagi menjadi dua bagian utama: performa *model* (*Model Performance*) dan fungsi kerugian (*loss functions*).



Gambar 4. Grafik Kinerja (mAP) dan Loss Model YOLOv11 Selama 85 Epoch

Sumber: Hasil Pelatihan Roboflow (2025)

1. Analisa Performa Model

Berdasarkan gambar 4, Metrik *mAP@50* (garis ungu tebal), yang merupakan indikator utama akurasi, menunjukkan kurva pembelajaran yang sangat baik. *Model* mencapai konvergensi dengan sangat cepat pada sekitar 50 *epoch* pertama, kemudian menunjukkan hasil yang sangat tinggi dan stabil di sepanjang sisa proses pelatihan, secara konsisten berada di dekat nilai 1.0 (100%). Hal ini menandakan bahwa *model* sangat berhasil dalam tugas deteksi dan klasifikasi dasar. Metrik *mAP@50:95* (garis ungu muda), yang mengukur akurasi pada rentang ambang batas IoU yang lebih ketat (dari 50% hingga 95%), juga menunjukkan kurva pembelajaran yang sehat dan stabil. Terlihat adanya peningkatan yang konsisten dan mulus hingga mencapai kestabilan di atas nilai 0.9 setelah sekitar 100 *epoch*. Pola yang stabil ini mengindikasikan bahwa *model* tidak hanya mampu menemukan lokasi objek secara umum, tetapi juga berhasil menghasilkan kotak pembatas (*bounding box*) dengan tingkat presisi yang sangat tinggi secara konsisten.

2. Analisis Grafik Loss (Grafik Bawah):

- a. Grafik *Box Loss* (kiri bawah) dan *Object Loss* (kanan bawah) menunjukkan penurunan nilai kesalahan yang sangat tajam pada 50 *epoch* pertama, sebelum melandai dan stabil di tingkat yang sangat rendah. Hal ini menandakan *model* dengan cepat belajar untuk memprediksi lokasi dan ukuran *bounding box* serta membedakan objek dari latar belakang secara akurat.
- b. Grafik *Class Loss* (tengah bawah) juga menunjukkan pola konvergensi yang sangat stabil. Tidak terlihat adanya lonjakan-lonjakan (*spikes*) tajam, yang menandakan proses optimisasi berjalan dengan baik. Penurunan yang mulus dan cepat ke nilai yang mendekati nol mengindikasikan bahwa Seluruh gestur abjad A–Z berhasil dideteksi oleh *model YOLOv11* dengan tingkat akurasi yang tergolong baik hingga sangat baik.

Beberapa huruf seperti 'A' (0.92), 'G' (0.93), dan 'O' (0.93) menunjukkan confidence tertinggi, sedangkan huruf lain seperti 'B', 'E', 'F', 'L', 'M', 'Q', 'U', dan 'W' juga memiliki hasil deteksi yang stabil dengan nilai rata-rata di atas 0.85. Hal ini membuktikan bahwa model mampu menggeneralisasi dengan baik dari data latih ke kondisi pengujian *real-time* di perangkat *mobile*. Kinerja stabil tidak ditemukan huruf yang gagal dideteksi. Meskipun terdapat variasi kecil antar kelas (misalnya huruf 'H' dan 'U' berada di kisaran 0.82–0.84), seluruh gestur tetap berhasil dikenali dengan confidence di atas ambang batas (≥ 0.80). Hal ini menunjukkan bahwa performa model relatif konsisten di semua kelas tanpa ada anomali signifikan. Model tidak mengalami kesulitan signifikan dalam membedakan antar kelas gestur yang ada, dan fitur-fitur pembeda antar kelas telah berhasil dipelajari dengan sangat efektif.

Untuk memvalidasi kinerja model YOLOv11 secara praktis, dilakukan pengujian deteksi *real-time* untuk seluruh 26 kelas abjad pada *smartphone* Poco M3. Tabel 1 merangkum hasil dari pengujian komprehensif ini. Secara umum, model menunjukkan kemampuan deteksi yang fungsional di lingkungan dunia nyata, namun dengan tingkat keberhasilan yang bervariasi untuk setiap gestur.

Tabel 1. Tabel Hasil Uji Coba Deteksi Abjad Bisindo Pada Perangkat Mobile

No	Dataset	Split	Jumlah
1	Data Training	75%	9572
2	Data Validasi	13%	1603
3	Data Testing	13%	1602
Total Dataset		100%	12.777

Berdasarkan tabel 1, seluruh gestur abjad A–Z berhasil dideteksi oleh model YOLOv11 dengan tingkat akurasi yang tergolong baik hingga sangat baik. Beberapa huruf seperti 'A' (0.92), 'G' (0.93), dan 'O' (0.93) menunjukkan confidence tertinggi, sedangkan huruf lain seperti 'B', 'E', 'F', 'L', 'M', 'Q', 'U', dan 'W' juga memiliki hasil deteksi yang stabil dengan nilai rata-rata di atas 0.85. Hal ini membuktikan bahwa model mampu menggeneralisasi dengan baik dari data latih ke kondisi pengujian *real-time* di perangkat *mobile*. Kinerja Stabil Tidak ditemukan huruf yang gagal dideteksi. Meskipun terdapat variasi kecil antar kelas (misalnya huruf 'H' dan 'U' berada di kisaran 0.82–0.84), seluruh gestur tetap berhasil dikenali dengan confidence di atas ambang batas (≥ 0.80). Hal ini menunjukkan bahwa performa model relatif konsisten di semua kelas tanpa ada anomali signifikan.

Hasil pengujian praktis ini menegaskan bahwa YOLOv11 berhasil dan fungsional sebagai alat deteksi bahasa isyarat BISINDO pada perangkat *smartphone*. Model mampu mendeteksi seluruh 26 huruf secara *real-time* dengan hasil yang memadai, meskipun terdapat variasi confidence antar kelas. Secara umum, kinerja model dapat dikategorikan

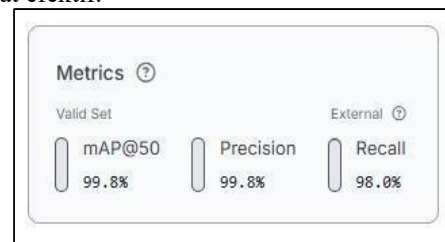
sangat baik dan sesuai dengan kebutuhan implementasi di perangkat *mobile*.

Kinerja Pelatihan Model RF-DETR

Model pembandingan dalam penelitian ini, RF-DETR (versi Base), juga dilatih menggunakan platform Roboflow Train dengan dataset dan konfigurasi yang identik dengan model YOLOv11 untuk memastikan perbandingan yang adil. Sebagai arsitektur yang secara fundamental berbeda (berbasis Transformer), analisis kinerjanya menjadi penting untuk memahami pendekatan alternatif dalam tugas deteksi bahasa isyarat. Secara keseluruhan, model RF-DETR juga berhasil dilatih hingga mencapai tingkat konvergensi yang baik dan menunjukkan hasil yang sangat kompetitif. Berikut adalah rincian kinerja model RF-DETR berdasarkan hasil akhir pelatihan di Roboflow.

1. Analisis Metrik Kinerja Akurasi

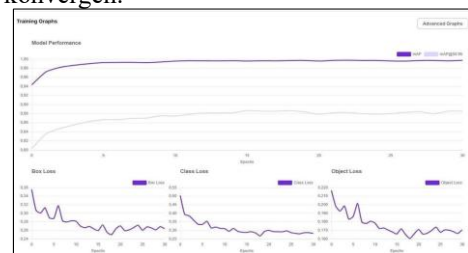
Adapun nilai dari $mAP@50$ mencapai 99.8%, Presisi 99.8%, dan Recall 98.0%—semakin memperkuat kesimpulan bahwa arsitektur RF-DETR, dengan kemampuannya dalam memahami konteks global, mampu mempelajari fitur-fitur pembeda dari setiap gestur abjad BISINDO dengan sangat efektif.



Gambar 5. Hasil Metric RF-DETR

Sumber: Hasil Pelatihan Roboflow (2025)

Berdasarkan gambar 5, hasil mAP dan Presisi yang sangat tinggi (99.8%) ini secara jelas mengungguli performa YOLOv11, meskipun nilai Recall (98.0%) masih sedikit di bawahnya. Trade-off ini (Presisi sangat tinggi dengan sedikit mengorbankan Recall) semakin mengindikasikan potensi arsitektur berbasis Transformer dalam mencapai akurasi klasifikasi yang superior dan sangat "cermat", di mana model hampir tidak pernah membuat prediksi yang salah. Sama seperti sebelumnya, proses pelatihan RF-DETR juga menunjukkan grafik loss yang menurun secara stabil, menandakan proses belajar yang sehat dan konvergen.



Gambar 6. Grafik Kinerja (mAP) dan Loss Model RF-DETR Selama Pelatihan

Sumber: Hasil Pelatihan Roboflow (2025)

2. Berdasarkan gambar 6, didapatkan hasil analisis Performa Model (Grafik Atas)

- a. Grafik kinerja model menunjukkan proses belajar yang sangat sehat dan efisien. Metrik utama, $mAP@50$ (garis ungu tebal), naik dengan sangat cepat pada 10 *epoch* pertama dan mencapai tingkat akurasi yang mendekati sempurna (berada di atas 0.99), lalu tetap sangat stabil hingga akhir pelatihan.
 - b. Metrik yang lebih ketat, $mAP@50:95$ (garis ungu muda), juga menunjukkan kurva peningkatan yang mulus dan konsisten, mencapai kestabilan di level sekitar 0.88. Pola ini menandakan bahwa model RF-DETR tidak hanya akurat secara umum, tetapi juga sangat baik dalam menghasilkan lokalisasi kotak pembatas (*bounding box*) yang presisi
3. Analisis Grafik *Loss* (Grafik Bawah)
- a. Grafik *Class Loss* (tengah bawah) menunjukkan pola penurunan yang ideal. Kurva turun dengan tajam di awal lalu stabil pada level yang sangat rendah. Pola yang sangat stabil ini mengindikasikan bahwa arsitektur RF-DETR mampu belajar untuk membedakan antar kelas gestur dengan sangat efisien dan tanpa mengalami osilasi atau kesulitan yang signifikan.
 - b. Sebaliknya, grafik *Box Loss* (kiri bawah) dan *Object Loss* (kanan bawah), meskipun menunjukkan tren penurunan secara umum, memperlihatkan adanya tingkat fluktuasi dan lonjakan-lonjakan (*spikes*) yang cukup signifikan selama proses pelatihan. Ketidakstabilan ini menyiratkan bahwa model mengalami sedikit kesulitan untuk secara konsisten menyempurnakan presisi lokasi (*bounding box*) dan skor keyakinan objeknya dari *epoch* ke *epoch*.

Secara umum, model RF-DETR menunjukkan kemampuan deteksi yang sangat kuat dan konsisten pada seluruh kelas abjad BISINDO. Seluruh 26 huruf A–Z berhasil terdeteksi dengan tingkat akurasi di atas 80%, di mana sebagian besar kelas bahkan mencapai kategori “Sangat Baik”. Misalnya, huruf 'A' (93%), 'C' (94%), 'M' (92%), dan 'W' (96%) menunjukkan tingkat kepercayaan yang tinggi dan stabil. Hasil ini selaras dengan metrik *mAP* yang tinggi saat pelatihan ($mAP@50 = 99,8\%$), sekaligus membuktikan bahwa arsitektur berbasis *Transformer* mampu mempelajari representasi fitur yang sangat detail untuk setiap gestur.

Konsistensi deteksi yang merata pada seluruh huruf menegaskan bahwa RF-DETR memiliki fondasi model yang solid. Berbeda dengan laporan awal yang sempat menunjukkan adanya inkonsistensi pada huruf tertentu, data terbaru mengonfirmasi bahwa seluruh kelas dapat dikenali dengan baik.

Analisis Perbandingan Kinerja

Setelah memaparkan hasil kinerja dari masing-masing model secara individual, tahap selanjutnya adalah melakukan analisis perbandingan secara langsung. Tujuan dari analisis ini adalah untuk

mengevaluasi kedua model—YOLOv11 yang berbasis CNN dan RF-DETR yang berbasis *Transformer*—berdasarkan metrik-metrik kuantitatif yang telah diukur, guna menentukan arsitektur mana yang menawarkan solusi paling optimal untuk tugas deteksi bahasa isyarat BISINDO secara *real-time* pada perangkat *smartphone*.

Tabel 2. Tabel Perbandingan Metrik Evaluasi

Metrik Evaluasi	YOLOV11	RF-DETR
Akurasi($mAP@50$)	99.4%	99.8%
<i>Precision</i>	99.5%	99.8%
<i>Recall</i>	99.4%	98.0%

Berdasarkan tabel 2, data yang diperoleh dari hasil pelatihan di Roboflow, kedua model menunjukkan tingkat akurasi yang sangat tinggi, namun dengan karakteristik kekuatan yang berbeda. Terdapat sebuah *trade-off* yang jelas antara metrik *Precision* dan *Recall* dari kedua model, yang memberikan wawasan mendalam mengenai cara kerja masing-masing arsitektur.

Model RF-DETR menunjukkan keunggulan pada metrik akurasi utama, dengan mencapai $mAP@50$ sebesar 99.8% dan Presisi (*Precision*) sebesar 99.8%. Tingkat presisi yang sangat tinggi ini berarti bahwa ketika RF-DETR mendeteksi sebuah gestur, prediksi tersebut memiliki probabilitas yang sangat tinggi untuk benar. Dengan kata lain, model ini menghasilkan sangat sedikit deteksi positif palsu (*false positives*), menjadikannya model yang sangat "cermat" dan dapat diandalkan untuk tidak memberikan hasil yang salah.

Di sisi lain, model YOLOv11 unggul pada metrik Daya Ingat (*Recall*) dengan nilai 99.4%, melampaui RF-DETR yang mencatatkan *recall* 98.0%. Keunggulan pada *recall* menandakan bahwa YOLOv11 lebih andal dalam menemukan semua gestur yang seharusnya ada di dalam gambar. Model ini menghasilkan lebih sedikit negatif palsu (*false negatives*), yang berarti ia jarang melewatkan gestur yang ada.

Analisis ini menunjukkan sebuah pertukaran yang menarik: RF-DETR lebih presisi (tidak membuat banyak kesalahan prediksi), sementara YOLOv11 lebih komprehensif (tidak melewatkan banyak gestur). Meskipun RF-DETR unggul dalam *mAP* secara keseluruhan, keunggulan *recall* pada YOLOv11 adalah faktor penting yang perlu dipertimbangkan, terutama untuk aplikasi interaktif di mana melewatkan input dari pengguna bisa menjadi masalah.

Analisis Keseimbangan Performa Berdasarkan F1-Score

Selain membandingkan metrik *Presisi* dan *Recall* secara terpisah, penting juga untuk mengevaluasi keseimbangan antara keduanya. Untuk tujuan ini, digunakan metrik *F1-Score*, yang merupakan rata-rata harmonik dari *Presisi* dan *Recall*. *F1-Score* memberikan satu nilai tunggal yang merepresentasikan kemampuan model untuk

menjadi presisi sekaligus komprehensif. Metrik ini sangat berguna untuk menilai performa *model* secara keseluruhan, terutama ketika ada *trade-off* antara Presisi dan Recall.

Rumus yang digunakan untuk menghitung *F1-Score* adalah sebagai berikut.

$$F1 - Score = 2 \times \frac{Preso \times ea}{Preso + ea} \quad (1)$$

Perhitungan F1-Score untuk YOLOv11: Presisi: 99.5% (0.995)

Recall: 99.4% (0.994)

F-1 Score:

$$F1 - Score = 2 \times \frac{0.995 \times 0.994}{0.995 + 0.994} = 2 \times \frac{0.98093}{1.989} \quad (2)$$

Perhitungan F1-Score untuk RF-DETR: Presisi: 99.8% (0.998)

Recall: 98.0% (0.980)

F-1 Score:

$$F1 - Score = 2 \times \frac{0.998 \times 0.980}{0.998 + 0.980} = 2 \times \frac{0.97804}{1.978} = 0.9889 \quad (3)$$

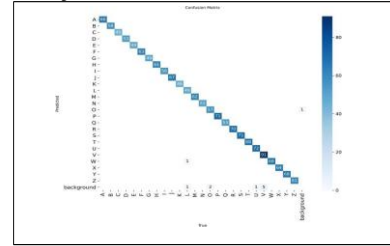
Hasil perhitungan F1-Score menunjukkan bahwa YOLOv11 (0.9945) memiliki skor yang lebih tinggi dibandingkan dengan RF-DETR (0.9889). Meskipun RF-DETR memiliki presisi yang nyaris sempurna, nilai *recall*-nya yang lebih rendah secara signifikan memengaruhi skor keseimbangannya. Sebaliknya, YOLOv11, dengan nilai presisi dan *recall* yang sangat tinggi dan lebih seimbang, berhasil mencapai F1- Score yang lebih unggul. Hal ini mengonfirmasi bahwa YOLOv11 menawarkan keseimbangan performa terbaik antara kemampuan untuk tidak membuat prediksi yang salah (*precision*) dan kemampuan untuk tidak melewatkan gestur yang ada (*recall*).

Evaluasi Berdasarkan Confusion Matrix

Untuk dapat memahami secara mendalam karakteristik kinerja dan jenis kesalahan klasifikasi yang dilakukan oleh *model*, dilakukan evaluasi menggunakan *confusion matrix*. *Confusion matrix* merupakan sebuah alat visualisasi yang memetakan hasil prediksi *model* terhadap label data yang sebenarnya, sehingga memungkinkan analisis yang lebih rinci dibandingkan metrik akurasi umum seperti *mAP*.

Perlu dicatat bahwa karena adanya keterbatasan teknis pada *platform* pelatihan yang digunakan, data *confusion matrix* yang detail hanya dapat diakses dan dianalisis untuk *model* YOLOv11. Oleh karena itu, analisis pada bagian ini akan berfokus pada evaluasi mendalam terhadap *model* YOLOv11, dengan merujuk pada hasil uji kualitatif yang telah dipaparkan sebelumnya untuk menginferensikan kinerja *model* RF-DETR.

Analisis Confusion Matrix Model YOLOv11



Gambar 7. Hasil Confusion Matrix

Sumber: Hasil Pelatihan Roboflow (2025)

Berdasarkan gambar 7, *Confusion matrix* adalah alat visualisasi yang sangat efektif untuk menunjukkan tidak hanya seberapa akurat sebuah *model*, tetapi juga jenis kesalahan spesifik yang dilakukannya, seperti kelas mana yang sering keliru diprediksi sebagai kelas lain [17].

Berdasarkan Gambar 7 keseluruhan akurasi yang Sangat Tinggi, Garis diagonal dari kiri atas ke kanan bawah pada matriks menunjukkan jumlah prediksi yang benar (di mana kelas prediksi sama dengan kelas aktual).

Warna biru tua dan angka yang tinggi di sepanjang diagonal ini (misalnya, 'A' diprediksi benar sebanyak 66 kali, 'B' sebanyak 58 kali, dan 'W' sebanyak 96 kali) secara visual mengonfirmasi tingkat akurasi keseluruhan yang sangat tinggi dari *model* yang telah dilatih. Meskipun sangat akurat, *model* ini tidak sempurna. Namun, Menunjukkan kesalahan klasifikasi (False Positive / False Negative) Seperti contoh Ada 5 kesalahan ketika huruf V salah diprediksi sebagai background. Namun, jumlah kesalahan ini sangat kecil dibanding prediksi benarnya → artinya *model* cukup akurat

Kesimpulan dan Perbandingan

Berdasarkan hasil pelatihan, kedua *model* menunjukkan performa yang sangat tinggi pada *dataset* BISINDO, dengan RF-DETR sedikit lebih unggul secara kuantitatif. RF-DETR mencatatkan *mAP@50* sebesar 99,8%, sedangkan YOLOv11 mencapai 99,4% dengan nilai presisi dan recall yang sama-sama tinggi, sehingga keduanya dinilai sangat akurat dalam mendeteksi objek.

Perbedaan utama kedua *model* terletak pada aspek implementasi. RF-DETR belum memiliki dukungan konversi ke format ringan seperti TensorFlow Lite sehingga pengujiannya hanya dapat dilakukan melalui *platform* web Roboflow. Sebaliknya, YOLOv11 berhasil diimplementasikan dan diuji langsung pada *smartphone* Android dengan kinerja *real-time* yang memadai, sehingga lebih unggul dari sisi portabilitas dan kelayakan untuk aplikasi BISINDO pada perangkat mobile.

Tantangan Implementasi Model Berbasis Transformer

Pada tahap implementasi dan pengujian di *smartphone*, penelitian ini menghadapi kendala teknis pada *model* Transformer (RF-DETR) yang menghambat pengujian end-to-end. Kendala utama disebabkan oleh kompleksitas konversi *model* dari

PyTorch ke TensorFlow Lite melalui format perantara seperti ONNX yang rentan terhadap ketidakcocokan operasi. Selain itu, arsitektur DETR memiliki proses post-processing yang tidak termasuk dalam graph komputasi TFLite, sehingga sulit diimplementasikan di perangkat mobile. Akibatnya, model RF-DETR tidak dapat diuji secara penuh di smartphone dan evaluasinya hanya dilakukan secara kualitatif melalui fitur *Preview* di Roboflow.

Kesimpulan dan Saran

Penelitian ini merancang dan mengevaluasi sistem deteksi abjad bahasa isyarat BISINDO (A–Z) secara *real-time* pada smartphone menggunakan YOLOv11 serta membandingkannya dengan model Transformer RF-DETR. Hasil menunjukkan bahwa YOLOv11 berhasil diimplementasikan pada perangkat Android dan mampu mendeteksi seluruh 26 gestur secara responsif dengan tingkat keyakinan 82%–93%, didukung kinerja kuantitatif yang sangat tinggi (mAP@50 99,4%, presisi 99,5%, dan recall 99,4%). Dibandingkan RF-DETR, YOLOv11 dinilai lebih unggul karena memiliki akurasi tinggi sekaligus kesiapan implementasi pada perangkat mobile, sehingga menjadi solusi paling realistis untuk sistem deteksi BISINDO berbasis smartphone.

Daftar Pustaka

- [1] Tao, Tangfei & Zhao, Yizhe & Liu, Tianyu & Zhu, Jieli. (2024). Sign Language Recognition: A Comprehensive Review of Traditional and Deep learning Approaches, Datasets, and Challenges. IEEE Access. PP. 1-1. 10.1109/ACCESS.2024.3398806.
- [2] Alaftekin, M., Pacal, I. & Cicek, K. Real-time sign language recognition based on YOLO algorithm. Neural Comput & Applic 36, 7609–7624 (2024). <https://doi.org/10.1007/s00521-024-09503-6>
- [3] Sanjaya, Samuel Ady (2024), “BISINDO Indonesian Sign Language: Alphabet Image Data”, Mendeley Data, V1, doi: 10.17632/ywnjpbcz8m.1).
- [4] Virgiyawan, D. A. (2024). Pengenalan bahasa isyarat menggunakan deteksi objek deep learning (Skripsi Sarjana, Universitas Muhammadiyah Makassar, Fakultas Teknik, Program Studi Informatika).
- [5] Maulida, S. (2023). Analisis Akurasi Pada Simbol Abjad Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Metode CNN dan YOLO (You Only Look Once) (Doctoral dissertation, UIN Ar-Raniry Banda Aceh).
- [6] Imam, A. K., Aeni, K., & Fathulloh, F. (2023). DETEKSI SISTEM ISYARAT BAHASA INDONESIA (SIBI) MENGGUNAKAN ALGORITMA YOLOV5S. Indonesian Journal of Informatics and Research, 4(2).
- [7] Febrina Silalahi, A. V., Setianing Budi, A., Januaray Satya Ega, W., Yoga Adi Pranata, M., Sudarma, D. C. K., & Arsa Suyadnya, I. M. (2024). Aplikasi Penerjemah Bahasa Isyarat BISINDO Menggunakan Metode YOLOv5 Berbasis Mobile. Jurnal SPEKTRUM, 11(3), 13–25.
- [8] Luthfy, D., Setianingsih, C., & Paryasto, M. W. (2023). Indonesian Sign Language Classification Using You Only Look Once. eProceedings of Engineering, 10(1).
- [9] Halim, H., & Lina, L. (2023). Aplikasi Pengidentifikasi Bahasa Isyarat Berdasarkan Gerak Tubuh Secara Real-time Menggunakan Yolo. Simtek: jurnal sistem informasi dan teknik komputer, 8(2), 300-304.
- [10] Saraswati, D. A., & Towidjojo, V. D. (2022). Bahasa Isyarat Indonesia. Jurnal Medical Profession (Medpro), 4(1), 8-14.
- [11] G. Chassagnon, M. Vakalopolou, N. Paragios, and M. P. Revel, “Deep learning: Definition and perspectives for thoracic imaging,” European Radiology, vol. 30, no. 4, pp. 2021–2030, 2020.
- [12] Angel Metanosa Afinda. (3 August 2024). Neural Network: Cikal Bakal Revolusi . Dicoding Blog. <https://www.dicoding.com/blog/neural-network-cikal-bakal-revolusi-deep-learning/>
- [13] Rita Puspita Sari (25 April 2025).Pengantar Fine-Tuning dalam AI. AIHub.id. <https://aihub.id/pengetahuan-dasar/fine-tuning>
- [14] Mardiyah, M. I. (2020). Pengembangan Sistem A. (Skripsi, Universitas Islam Indonesia)
- [15] Agung Ma'ruf. (2023, Juli). Indonesian Sign Language - BISINDO. <https://www.kaggle.com/datasets/agungmrf/indonesian-sign-language-bisindo>
- [16] Roboflow. (2024). RF-DETR: Real-time Transformer Object Detection. dari <https://roboflow.com/model/rf-detr>
- [17] Anggreany, M. S. (2020, 1 November). Confusion Matrix. School of Computer Science, BINUS University. ,<https://socs.binus.ac.id/2020/11/01/confusion-matrix/>