SKANIKA: Sistem Komputer dan Teknik Informatika
Volume 9, Nomor 2, Juli 2026, Halaman 320-333 E-ISSN: 2721-4788

Penerapan YOLOv13 untuk Deteksi Alfabet Bahasa Isyarat Indonesia secara Real-Time

Muhammad Noval Rais[1*], Sawali Wahyu[2]
[1,2]Ilmu Komputer, Teknik Informatika, Universitas Esa Unggul, Jakarta, Indonesia
E-mail: [1*]vallrais23@student.esaunggul.ac.id, [2]sawaliwahyu@esaunggul.ac.id
(* : corresponding author)

Abstrak

Bahasa Isyarat Indonesia (BISINDO) merupakan sarana komunikasi utama komunitas Tuli. Namun, rendahnya pemahaman masyarakat serta kesulitan dalam pengenalan gestur serupa dan adaptasi pencahayaan dinamis masih menjadi hambatan. Penelitian ini menerapkan arsitektur YOLOv13-Nano dengan modul HyperACE dan FullPAD untuk deteksi abjad statis BISINDO (A–Z). Model dilatih menggunakan 5.835 citra Mendeley Dataset (rasio 70:15:15) dengan pelabelan otomatis MediaPipe Hands, serta diimplementasikan pada dashboard FastAPI dengan logika debouncing dan temporal *auto-spacing*. Pengujian *offline* menunjukkan YOLOv13-Nano mencapai akurasi 89,22%, mAP50 83,02%, Precision 90,63%, Recall 89,62%, *F1-Score* 89,74%, dan waktu inferensi 23,62 ms. Model ini mengungguli baseline K-Nearest Neighbors (KNN), di mana KNN (*Full Image*) hanya mencapai akurasi 9,90% dan KNN (*Hand Cropped*) mencapai 11,23%. Pengujian *real-time* terbatas pada subjek uji menunjukkan Word Accuracy 87,4% pada kondisi normal, 88,3% pada *low-light* ekstrem, serta akurasi abjad 90,4% pada latar belakang kompleks. Hasil penelitian ini menunjukkan potensi YOLOv13-Nano untuk mendukung sistem penerjemah BISINDO *real-time* yang responsif pada beberapa kondisi.

Kata kunci: BISINDO, FullPAD, HyperAce, Real-Time, YOLOv13

Abstract

Indonesian Sign Language (BISINDO) is the primary communication medium for the deaf community. However, public understanding remains limited, and recognizing similar gestures and adapting to dynamic lighting pose significant challenges. This study applies the YOLOv13-Nano architecture with HyperACE and FullPAD modules for static BISINDO alphabet detection (A–Z). The model was trained on 5,835 images from the Mendeley Dataset (70:15:15 split) using MediaPipe Hands auto-labeling, and implemented on a FastAPI dashboard with debouncing and temporal auto-spacing logics. Offline testing shows YOLOv13-Nano achieved 89.22% classification accuracy, 83.02% mAP50, 90.63% Precision, 89.62% Recall, 89.74% F1-Score, and 23.62 ms inference time. This model outperforms K-Nearest Neighbors (KNN) baselines, where KNN (Full Image) achieved 9.90% accuracy and KNN (Hand Cropped) achieved 11.23% accuracy. Limited real-time testing on the test subject yielded a Word Accuracy of 87.4% under normal conditions, 88.3% under extreme low-light, and 90.4% alphabet accuracy against complex backgrounds. These results demonstrate the potential of YOLOv13-Nano to support a responsive real-time BISINDO translation system under various real-world environmental conditions.

Keywords*: BISINDO, FullPAD, HyperAce, YOLOv13, Real-Time.*

1. PENDAHULUAN

Bahasa Isyarat Indonesia (BISINDO) merupakan instrumen komunikasi utama bagi komunitas Tuli di Indonesia [1], yang jumlahnya tercatat mencapai lebih dari 22,97 juta jiwa penyandang disabilitas berdasarkan data Survei Sosial Ekonomi Nasional 2022 [2]. Meskipun populasinya signifikan, interaksi antara komunitas Tuli dan masyarakat umum masih terkendala oleh rendahnya pemahaman terhadap BISINDO [3], sehingga kebutuhan akan teknologi penerjemah otomatis menjadi semakin mendesak. Perkembangan *computer vision* berbasis *deep learning*, khususnya algoritma *You Only Look Once* (YOLO), telah membuka peluang besar untuk membangun sistem deteksi gestur tangan secara *real-time* dengan tingkat akurasi tinggi [4].

Namun demikian, sejumlah isu teknis masih membayangi implementasinya di dunia nyata, terutama pada aspek kemampuan model membedakan gestur yang secara visual mirip serta ketahanannya terhadap variasi kondisi lingkungan seperti pencahayaan yang tidak seragam dan latar belakang yang kompleks. Metode ini telah sukses diterapkan pada berbagai skenario deteksi objek nyata, salah satunya pada sistem penghitungan kendaraan [5].

Sejumlah penelitian terdahulu telah mengupayakan penyelesaian permasalahan tersebut dengan pendekatan yang beragam. Farhana et al. [6] menerapkan kombinasi YOLOv8 dan CNN untuk deteksi alfabet BISINDO dan memperoleh akurasi 89,74% pada dataset uji mereka, namun penelitian tersebut dibangun di atas volume dataset yang sangat terbatas, yaitu hanya 417 citra untuk 26 kelas huruf, sehingga variasi fitur yang dipelajari model menjadi minim dan berpotensi menurunkan kemampuan generalisasi pada kondisi baru. Pada sisi lain, Renaningtias et al. [7] mengimplementasikan YOLOv7 dan mencatatkan metrik evaluasi *offline* yang tinggi (mAP 0,995; Precision dan Recall 1,00) pada dataset mereka, tetapi melaporkan adanya penurunan akurasi signifikan pada pengujian *real-time* di kondisi pencahayaan rendah dan pergerakan objek yang cepat. Temuan serupa juga dilaporkan oleh Azahra [8] yang menggunakan YOLOv11 dan mencatatkan akurasi pelatihan sebesar 97% pada data mereka, meskipun model diidentifikasi masih kesulitan mengenali perbedaan spasial halus (*fine-grained spatial distinctions*) antar-kelas huruf yang mirip secara visual, seperti pasangan huruf 'M' dan 'N' atau 'K' dan 'P'.

Keterbatasan volume dataset juga ditemukan pada penelitian Munandar et al. [9] yang menggunakan YOLOv5 dengan 3.900 citra (150 citra per kelas) dan memperoleh *F1-score* 87,2%. Sementara itu, penelitian Daniels et al. [10] yang menerapkan YOLOv3 dan memperoleh akurasi 100% pada pengujian citra statis mereka, namun mendokumentasikan penurunan akurasi menjadi sekitar 73% ketika diuji pada video *real-time*, khususnya pada kondisi pencahayaan rendah. Di sisi lain, penelitian Ariansyah [11] yang menggunakan YOLOv8 dengan dataset lebih besar (25.000 gambar) berhasil mencapai akurasi 93,8% pada deteksi kata bahasa isyarat, tetapi data yang digunakan hanya bersumber dari foto *smartphone* tanpa variasi isyarat dari individu berbeda, sehingga generalisasi model terhadap keragaman pengguna di dunia nyata masih memerlukan validasi lebih lanjut.

Perkembangan arsitektur YOLO generasi terbaru turut memperkuat justifikasi pemilihan metode pada penelitian ini. Studi *benchmarking* oleh Gao et al. [12] yang membandingkan YOLOv8 hingga YOLOv13 pada konteks interaksi manusia-robot menunjukkan bahwa YOLOv13 unggul dalam ketahanan terhadap *noise* dan kondisi pencahayaan buruk dengan capaian mAP 0,990, meskipun pengujian tersebut masih menggunakan dataset berskala kecil. Sejalan dengan itu, Lei et al. [13] selaku pengusul arsitektur YOLOv13 menegaskan bahwa modul HyperACE (*Hypergraph-based Adaptive Correlation Enhancement*) memberikan keseimbangan antara akurasi dan efisiensi komputasi melalui ekstraksi korelasi tingkat tinggi antar-piksel, namun pengujian arsitektur tersebut masih terbatas pada deteksi objek umum sehingga penerapannya pada domain spesifik seperti BISINDO memerlukan evaluasi lebih lanjut.

Berdasarkan tinjauan terhadap penelitian-penelitian tersebut, dapat diidentifikasi tiga celah utama (*research gap*) yang belum terselesaikan secara simultan: (1) keterbatasan volume dan variasi dataset yang membatasi generalisasi model, (2) kesenjangan performa antara evaluasi *offline* yang tinggi dengan ketahanan *real-time* yang rendah, khususnya pada kondisi pencahayaan yang tidak ideal, dan (3) minimnya penelitian yang mengevaluasi arsitektur YOLOv13 secara spesifik pada domain deteksi BISINDO dengan pengukuran ketahanan lingkungan yang komprehensif.

Perlu dicatat bahwa nilai akurasi, presisi, mAP, *F1-score*, dan waktu inferensi yang dilaporkan dalam berbagai penelitian terdahulu di atas diperoleh menggunakan dataset, variasi lingkungan, dan spesifikasi perangkat keras yang berbeda. Oleh karena itu, metrik-metrik tersebut tidak dapat dibandingkan secara langsung sebagai ukuran keunggulan mutlak suatu model. Atas dasar tersebut, kebaruan dan kontribusi utama penelitian ini diletakkan pada penerapan serta evaluasi empiris arsitektur YOLOv13-Nano yang secara arsitektural memanfaatkan efisiensi modul FullPAD dan HyperACE pada domain deteksi alfabet statis BISINDO (A–Z) menggunakan subset Mendeley Dataset yang lebih masif (5.835 citra). Penelitian ini berfokus

pada pengujian ketahanan lingkungan secara langsung dan terukur pada skenario pencahayaan ekstrem redup (*low-light*) serta latar belakang kompleks (*cluttered background*). Selain mengukur ketahanan secara *real-time* melalui indikator Akurasi Kata (*Word Accuracy*) dan Waktu Inferensi (*Inference Time*), penelitian ini juga menyediakan perbandingan langsung (*head-to-head*) dengan algoritma baseline non-blackbox *K-Nearest Neighbors* (KNN) pada dataset dan perangkat keras yang sama untuk menjamin validitas evaluasi.

2. METODE PENELITIAN

Bab ini menguraikan secara rinci metode yang digunakan dalam penelitian pengembangan sistem deteksi Bahasa Isyarat Indonesia (BISINDO) berbasis arsitektur YOLOv13. Pembahasan diawali dengan desain penelitian yang menjadi kerangka dasar pendekatan yang diambil, dilanjutkan dengan penjelasan objek dan sumber data, teknik pengumpulan data, tahapan penelitian secara menyeluruh, proses pra-pemrosesan data, rancangan arsitektur model yang diusulkan, implementasi sistem deteksi secara *real-time*, hingga metrik evaluasi yang digunakan untuk mengukur keberhasilan model. Uraian pada bab ini disusun secara sistematis agar tahapan penelitian dapat ditelusuri dan direplikasi secara utuh, mulai dari perancangan hingga pengujian sistem.

2.1 Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimental kuantitatif dalam bidang *computer vision*, dengan fokus pada pengembangan dan pengujian model deteksi objek untuk mengenali abjad statis Bahasa Isyarat Indonesia (BISINDO). Objek yang diteliti adalah gestur tangan yang merepresentasikan 26 huruf alfabet (A–Z), dengan solusi yang diusulkan berupa arsitektur YOLOv13 (*You Only Look Once*) yang dilatih untuk mendeteksi dan mengklasifikasikan gestur tersebut secara *real-time*.

Pendekatan ini dipilih karena permasalahan deteksi BISINDO bersifat *object detection* sistem tidak hanya perlu mengenali kelas huruf, tetapi juga menentukan lokasi tangan pada bingkai gambar (*bounding box*) secara presisi dan cepat, sehingga cocok diselesaikan dengan keluarga algoritma YOLO yang telah teruji unggul dalam kecepatan inferensi dan akurasi deteksi objek.

2.2 Objek dan Sumber Data Penelitian

Objek material penelitian adalah gestur tangan abjad statis BISINDO. Data citra yang digunakan bersumber dari Mendeley Dataset [14], sebuah repositori data terbuka (*open-source*). Penelitian ini secara khusus mengekstraksi subset *Original Images* untuk menjaga kemurnian struktur spasial tangan dan menghindari distorsi rasio aspek akibat resize paksa. Subset ini terdiri atas 5.835 citra yang telah terpetakan ke dalam 26 kelas huruf (A–Z).

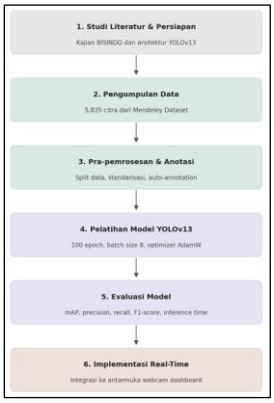
2.3 Teknik Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan melalui dua pendekatan yang saling melengkapi. Pendekatan pertama adalah pengumpulan data sekunder, yaitu dataset citra yang diunduh langsung dari repositori Mendeley Data, sehingga terhindar dari bias visual yang berpotensi muncul apabila data direkam secara mandiri (*data primer*). Pendekatan kedua adalah studi kepustakaan, berupa penelusuran literatur ilmiah mengenai struktur linguistik BISINDO, evolusi arsitektur YOLO, serta solusi komputasional atas masalah misklasifikasi gestur yang mirip secara visual, misalnya huruf ‘M’ dengan ‘N’, atau ‘K’ dengan ‘P’, yang menjadi landasan teoretis dalam merancang parameter model.

2.4 Tahapan Penelitian

Penelitian dilaksanakan melalui enam tahapan berurutan sebagaimana diilustrasikan pada Gambar 1. Tahapan pertama adalah studi literatur dan persiapan mengenai penerapan arsitektur YOLOv13 dan struktur alfabet statis BISINDO. Tahapan kedua adalah pengumpulan data, yaitu ekstraksi data mentah dari direktori *Original Images* pada repositori Mendeley yang langsung didistribusikan secara acak ke dalam data latih 70%, validasi 15%, dan uji 15% sejak awal guna

mencegah kebocoran data. Tahapan ketiga adalah pra-pemrosesan dan anotasi, mencakup standarisasi resolusi dan *auto-annotation*, dengan augmentasi dinamis serta sintesis *Mosaic* dan *Mixup* khusus pada data latih. Tahapan keempat adalah pelatihan model YOLOv13 1 menggunakan 100 *epoch*, *batch size* 8, *optimizer AdamW*, dan resolusi masukan 640×640 piksel. Tahapan kelima adalah evaluasi model melalui metrik kuantitatif (*mAP, Precision, Recall, F1-Score, Inference Time*) dan analisis kualitatif *Confusion Matrix*. Tahapan keenam adalah implementasi *real-time*, yaitu integrasi bobot model terbaik ke antarmuka kamera *webcam* untuk pengujian langsung di lingkungan nyata.



Gambar 1. Tahapan Penelitian Deteksi BISINDO dengan YOLOv13

2.5 Pra-Pemrosesan Data dan Konfigurasi Pelatihan

Sebelum data diumpankan ke jaringan saraf tiruan, dataset mentah melewati tahapan pra-pemrosesan menggunakan Python yang meliputi: ekstraksi data mentah dari 26 sub-direktori kelas, pemisahan data secara acak menjadi data *train* 70% (4.084 citra), *validation* 15% (875 citra), dan *test* 15% (876 citra) terlihat pada Tabel 1, augmentasi dasar pada data latih; standardisasi dimensi seluruh citra menjadi 640×640 piksel; serta *auto-annotation* menggunakan MediaPipe Hands [15].

Tabel 1. Proporsi Pembagian Dataset YOLOv13

Kategori Data	Jumlah Citra	Persentase
Training	4.084	70 %
Validation	875	15 %
Testing	876	15 %
Total	5.835	100 %

Tahapan *auto-annotation* diimplementasikan dengan melacak 21 titik koordinat (*landmarks*) tangan secara programatis, mengambil nilai ekstrem koordinat, lalu mengonversinya ke format standar YOLO dengan injeksi margin dinamis sebesar 15% untuk mencegah terpotongnya anatomi jari. Untuk memvalidasi hasil anotasi otomatis ini, seluruh kotak pembatas (*bounding box*) disaring secara visual. Hasil penyaringan menunjukkan bahwa MediaPipe Hands memiliki tingkat kegagalan deteksi sebesar 15,60% (910 dari total 5.835 citra), yang umumnya terjadi akibat variasi rotasi tangan ekstrem atau saturasi warna kulit yang rendah. Citra-citra yang gagal dideteksi secara otomatis ini kemudian diperbaiki secara manual menggunakan perangkat lunak anotasi *LabelImg* untuk menggambar ulang kotak pembatas secara presisi. Langkah validasi dan koreksi manual ini menjamin tingkat akurasi *ground truth* anotasi sebesar 100% sebelum dataset dibagi. Logika pemrograman dari fungsi *auto-annotation* ini ditunjukkan pada *code* berikut.

```
def get_hand_bboxes(img_rgb):
    results = hands_detector.process(img_rgb)
    if not results.multi_hand_landmarks:
        return []
```

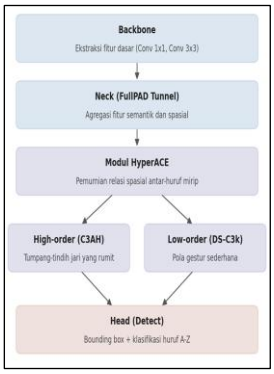
```
    bboxes = []
    for hand_landmarks in results.multi_hand_landmarks:
        # 1. Ekstrak koordinat x dan y dari 21 titik sendi jari
        x_coords = [lm.x for lm in hand_landmarks.landmark]
        y_coords = [lm.y for lm in hand_landmarks.landmark]

        # 2. Cari titik terluar (ekstrem)
        x_min, x_max = min(x_coords), max(x_coords)
        y_min, y_max = min(y_coords), max(y_coords)

        w = x_max - x_min
        h = y_max - y_min

        # 3. Injeksi margin 15% agar bentuk geometri jari tidak terpotong
        margin = 0.15
        x_min_m = max(0.0, x_min - w * margin)
        x_max_m = min(1.0, x_max + w * margin)
        y_min_m = max(0.0, y_min - h * margin)
        y_max_m = min(1.0, y_max + h * margin)

        # 4. Konversi ke matriks standar YOLO
        x_center = (x_min_m + x_max_m) / 2.0
        y_center = (y_min_m + y_max_m) / 2.0
        width = x_max_m - x_min_m
        height = y_max_m - y_min_m

        bboxes.append((x_center, y_center, width, height))
    return bboxes
```

Setelah proses anotasi divalidasi dan diperbaiki, proses augmentasi tingkat lanjut dilakukan secara *offline* pada porsi data latih untuk melipatgandakan variabilitas data hingga mencapai 11.324 objek label melalui teknik Mosaic dan Mixup. Pada porsi 20% data latih, augmentasi Mosaic diterapkan dengan menggabungkan empat citra latih acak yang diubah ukurannya menjadi 320×320 piksel ke dalam satu kanvas berukuran 640×640 piksel, dengan penyesuaian otomatis koordinat kotak pembatas sesuai posisi kuadran masing-masing gambar. Sementara itu, augmentasi Mixup juga diterapkan pada porsi 20% data latih lainnya dengan mencampur dua citra secara linear menggunakan rasio pencampuran (λ) acak antara 0,4 hingga 0,6 berdasarkan persamaan $L_{mix} = \lambda L_1 + (1 - \lambda)L_2$ di mana label koordinat dari kedua gambar asli digabungkan secara kumulatif. Selama proses pelatihan model YOLOv13, augmentasi *mosaic* dan *mixup* internal pada konfigurasi bawaan dinonaktifkan (mosaic=0.0, mixup=0.0) untuk menghindari augmentasi ganda pada dataset yang telah diproses secara *offline*. Meskipun demikian, augmentasi dinamis ringan berupa rotasi acak (±15°), translasi (10%), dan penskalaan (15%) tetap diaktifkan secara *online* untuk melatih fleksibilitas dan variabilitas posisi tangan subjek.

Model diinisialisasi menggunakan bobot prapelatihan (*pretrained weights*) yolov13n.pt yang dilatih pada dataset COCO untuk menerapkan *transfer learning*. Proses pelatihan dikonfigurasi menggunakan optimizer AdamW dengan initial *learning rate* sebesar 0,001 dan *weight decay* sebesar 0,001 untuk meminimalisasi risiko *overfitting* terhadap detail latar belakang. Pelatihan dijadwalkan berjalan selama 100 epoch dengan mengaktifkan parameter patience=50 sebagai mekanisme penahanan pelatihan (*early stopping*). Pelatihan akan dihentikan secara otomatis jika nilai mAP50 pada data validasi tidak menunjukkan peningkatan selama 50 epoch berturut-turut, namun model berhasil menyelesaikan seluruh 100 epoch karena metrik terus menunjukkan tren peningkatan yang positif. Evaluasi offline pada data uji menggunakan ambang batas keyakinan (confidence threshold) sebesar 0,25. Seluruh proses pelatihan dan evaluasi model ini dijalankan pada workstation lokal dengan spesifikasi berupa CPU AMD Ryzen 5 7640HS (6 Cores, 12 Threads, dengan kecepatan hingga 5,0 GHz), RAM 16 GB DDR5, serta akselerasi GPU NVIDIA GeForce RTX 3050 Laptop GPU dengan memori grafis (VRAM) sebesar 6 GB.

2.6 Model YOLOv13 yang Diusulkan

Model deteksi dibangun secara *end-to-end* menggunakan arsitektur YOLOv13, yang secara struktural terbagi menjadi empat komponen utama penyusun sebagaimana ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur YOLOv13 untuk Deteksi BISINDO

Secara ringkas: *Backbone* mengekstraksi tekstur dan kontur jari melalui operasi konvolusi; Neck dengan mekanisme FullPAD Tunnel memastikan informasi semantik dan spasial terdistribusi ke seluruh lapisan jaringan; modul HyperACE memecah aliran fitur menjadi cabang *high-order* (C3AH) dan *low-order* (DS-C3k) guna mendiskriminasi huruf-huruf yang mirip secara visual; dan Head menghasilkan koordinat *bounding box* serta probabilitas klasifikasi kelas huruf.

2.7 Implementasi Sistem Deteksi *Real-Time*

Model terbaik hasil pelatihan (yolov13n.pt versi termodifikasi) diintegrasikan ke dalam *pipeline* perangkat lunak berbasis FastAPI untuk streaming video secara asinkron [16]. Keluaran mentah model kemudian melewati dua tahapan pascapemrosesan (*post-processing*). Tahap pertama adalah filter kepercayaan dengan batas ambang batas keyakinan (*confidence threshold*) sebesar 0,75 untuk meminimalkan *false positives* akibat gangguan latar belakang. Tahap kedua adalah logika temporal dengan ukuran buffer 15 frame (0,5 detik pada kecepatan 30 FPS) untuk proses *debouncing* (mencegah fluktuasi deteksi huruf), serta batas hilangnya deteksi tangan selama 30 frame (1 detik) untuk menyisipkan spasi otomatis [17].

Untuk menguji performa sistem ini pada kondisi nyata secara terstandarisasi, protokol pengujian dilakukan secara terkontrol dengan melibatkan satu subjek uji (operator). Pengujian fungsionalitas dijalankan secara acak untuk 26 huruf alfabet statis BISINDO (A–Z) dengan 10 kali pengulangan untuk setiap huruf, serta 5 kali pengulangan untuk skenario pengujian pembentukan kata (ejaan kata "MAKAN AYAM"). Perekaman video *real-time* dilakukan menggunakan kamera web eksternal Logitech C270 HD yang menghasilkan resolusi 720p pada kecepatan frame rate 30 FPS, dengan jarak antara lensa kamera dan tangan subjek diatur secara konsisten pada rentang 40 cm hingga 60 cm. Proses inferensi *real-time* dijalankan pada laptop berspesifikasi prosesor AMD Ryzen 5 7640HS, RAM 16 GB, serta akselerasi grafis GPU NVIDIA GeForce RTX 3050 Laptop GPU (6 GB VRAM) dengan ambang batas deteksi (*confidence threshold*) ditingkatkan menjadi 0,75 guna mereduksi noise latar belakang. Pengujian ketahanan sistem dievaluasi pada tiga skenario kondisi lingkungan, yaitu kondisi normal dengan intensitas cahaya stabil berkisar 300 lux hingga 400 lux, kondisi pencahayaan rendah ekstrem (*extreme low-light*) di bawah 15 lux yang diukur dengan lux meter, serta kondisi latar belakang kompleks (*cluttered background*) yang dipenuhi oleh objek-objek mati bertekstur dan berwarna menyerupai warna kulit manusia seperti perabot kayu.

2.8 Metrik Evaluasi

Performa model diukur melalui mekanisme evaluasi dua lapis. Lapis pertama adalah evaluasi *offline* berbasis data uji untuk mengukur kapabilitas klasifikasi model menggunakan

standar ukur kerangka YOLO (YOLO *framework*) [18], yang secara komprehensif mencakup metrik *mAP*, *Precision*, *Recall*, dan *F1-score* yang lazim digunakan dalam tinjauan deteksi objek sedekade terakhir [19], serta *Confusion Matrix* [20]. Lapis kedua adalah evaluasi *real-time* yang mengukur ketahanan (*robustness*) sistem di lingkungan nyata, mencakup Akurasi Kata (*Word Accuracy*) dan Waktu Inferensi (*Inference Time*) dalam milidetik. Kombinasi kedua lapis evaluasi ini memberikan tolok ukur komprehensif dalam membuktikan efektivitas arsitektur YOLOv13 untuk deteksi BISINDO.

2.9 Klasifikasi Pembanding *Non-Blackbox* (KNN)

Guna mengukur tingkat kesulitan dataset citra dan menyediakan nilai dasar (*baseline*) pembanding yang tidak bersifat *blackbox*, penelitian ini turut mengimplementasikan algoritma K-Nearest Neighbors (KNN). Algoritma ini dipilih karena sifat pengambil keputusannya yang transparan berdasarkan kedekatan jarak spasial antar data latih. Selain itu, algoritma KNN juga telah banyak diterapkan dalam berbagai penelitian klasifikasi citra digital berbasis fitur visual [21]. Pada penelitian ini, evaluasi KNN dibagi menjadi dos skenario pengujian untuk menjamin pembandingan yang objektif (*fair baseline*). Skenario pertama adalah KNN (Full Image), di mana citra masukan diubah ke format grayscale dan diperkecil ukurannya menjadi 64x64 piksel untuk memperingan beban komputasi. Selanjutnya, dilakukan proses ekstraksi vektor (*flattening*) dengan merentangkan matriks piksel 2D menjadi susunan vektor 1D yang menghasilkan 4.096 titik fitur per citra. Skenario kedua adalah KNN (Hand Cropped), di mana area tangan dipotong terlebih dahulu (*cropped*) berdasarkan koordinat pembatas (*bounding box*) tangan dari MediaPipe, kemudian di-resize menjadi 64x64 piksel dan di-flatten. Skenario kedua ini bertujuan untuk melihat efektivitas klasifikasi KNN apabila faktor gangguan latar belakang (*background noise*) telah dihilangkan secara teoritis. Berdasarkan vektor tersebut pada masing-masing skenario, sistem menghitung jarak Euclidean antara citra uji dengan seluruh citra pada data latih. Pada tahap akhir penentuan kelas, parameter pencarian (K) diatur pada nilai 5, di mana sistem akan mencari 5 citra terdekat dan menetapkan prediksi klasifikasi alfabet berdasarkan probabilitas kemunculan label terbanyak (*majority voting*).

3. HASIL DAN PEMBAHASAN

Pembahasan pada bagian ini difokuskan pada realisasi dari metode yang diusulkan, mulai dari tahapan pemrosesan data, evaluasi pelatihan, konstruksi aplikasi, hingga pengujian ketahanan sistem di lingkungan nyata.

3.1 Implementasi Pra-Pemrosesan (*Pre-Processing*)

Sebelum memasuki fase pelatihan jaringan saraf tiruan, keseluruhan dataset dieksekusi secara terotomatisasi menggunakan skrip Python melalui sebuah *pipeline* pra-pemrosesan berjenjang. Rincian tahapan implementasi dari keseluruhan proses ini diuraikan pada Tabel 2 berikut.

Tabel 2. Alur Implementasi Pra-Pemrosesan Data

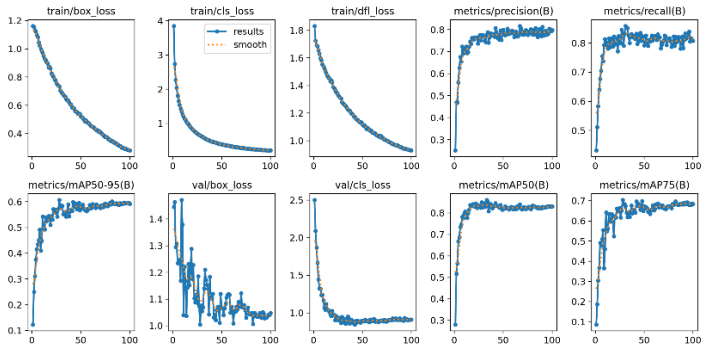
Tahap	Proses Implementasi	Luaran (*Output*) Sistem
1	Ekstraksi Data Mentah	5.835 citra *Original Images* dimuat ke dalam memori komputasi.
2	Pemisahan Data (*Split*)	Data dipartisi acak: *Train* 70%, *Validation* 15%, dan *Test* 15%.
3	Augmentasi Dasar	Injeksi variasi kecerahan, kontras, translasi, dan rotasi statis (±15°).
4	Standarisasi Dimensi	Matriks resolusi citra disamakan mutlak menjadi 640×640 piksel.
5	*Auto-Annotation*	Pemetaan titik koordinat kerangka tangan via *MediaPipe Hands*
6	Konversi Format YOLO	Koordinat dikonversi presisi menjadi *Bounding Box* berformat .txt.
7	Augmentasi Lanjutan	Sintesis hibrida *Mosaic* & *Mixup* khusus pada direktori latih.

Berdasarkan Tabel 2 di atas, implementasi *auto-annotation* menggunakan *MediaPipe* terbukti berhasil mengekstrak titik kerangka tangan menjadi batasan *bounding box* secara presisi. Selanjutnya, implementasi teknik augmentasi lanjutan pada Tahap 7 secara empiris berhasil mendongkrak kepadatan objek secara artifisial, menghasilkan lonjakan dari 5.835 citra menjadi

total 11.324 objek label. Pendekatan ini sukses menyuntikkan variabilitas visual yang sangat masif guna mencegah terjadinya *overfitting* selama proses pelatihan.

3.2 Hasil Pelatihan Model

Model YOLOv13 dilatih selama 100 epoch menggunakan *batch size* 8 dan optimizer AdamW. Kurva pelatihan menunjukkan tren penurunan *loss* yang konsisten, sementara metrik performa menunjukkan tren pendakian yang stabil. Keseimbangan antara metrik Precision (90,63%) dan Recall (89,62%) pada data uji menunjukkan bahwa model memiliki performa yang relatif seimbang dalam mendeteksi dan mengklasifikasikan kelas alfabet, yang di tunjukkan pada Gambar 3 berikut.

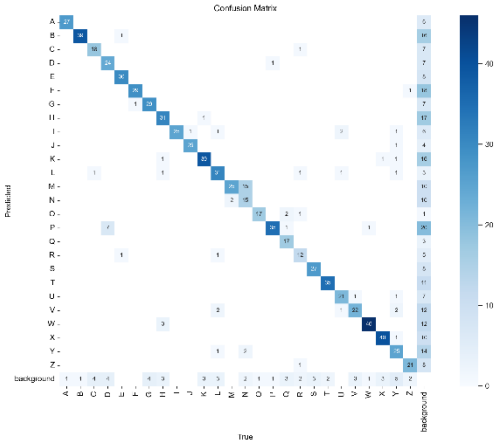


Gambar 3. Kurva Dinamika Pelatihan Model YOLOv13 (Loss dan mAP) Selama 100 Epoch

Meskipun demikian, analisis *Confusion Matrix* mengungkap adanya pola kesalahan klasifikasi yang signifikan pada beberapa pasangan gestur yang memiliki kemiripan geometri spasial tinggi. Pola kesalahan tertinggi ditemukan pada klasifikasi huruf N menjadi M dengan tingkat kesalahan mencapai 44%. Hal ini disebabkan oleh kepadatan spasial yang 95% identik antara gestur huruf 'M' dan 'N' pada proyeksi 2D (keduanya melibatkan penekukan jari ke arah telapak tangan dengan perbedaan posisi ibu jari yang sangat tipis). Degradasi performa ini mengonfirmasi keterbatasan alami kamera 2D dalam menangkap kedalaman informasi (*depth*), sehingga diperlukan integrasi sensor kedalaman 3D pada penelitian selanjutnya untuk mendiskriminasi gestur serupa secara lebih akurat. Rincian analisis misklasifikasi ini diuraikan pada Tabel 3 dan visualisasi confusion matrix hasil pelatihan YOLOv13 terlihat pada Gambar 4.

Tabel 3. Analisis Pola Misklasifikasi pada Confusion Matrix

Pasangan Abjad	***Error Rate***	***Root Cause***
N → M	44%	Kepadatan spasial 95% identik (*skin-on-skin*).
D → P	20%	Ambiguitas orientasi sudut dan kedalaman citra 2D.
Y, R, Q	11% - 13%	Dimensi *bounding box* terlalu tipis & rentan *motion blur*.

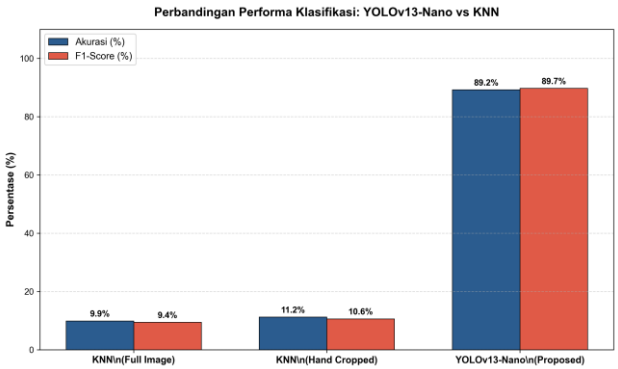


Gambar 4. Confusion Matrix Hasil Pelatihan Model YOLOv13

Untuk mengevaluasi kontribusi arsitektur YOLOv13-Nano yang diusulkan secara objektif, dilakukan pengujian komparatif *head-to-head* terhadap algoritma klasifikasi tradisional K-Nearest Neighbors (KNN) pada dataset pengujian yang sama (677 citra). Hasil perbandingan metrik disajikan pada Tabel 4, sedangkan visualisasi grafik perbandingannya ditunjukkan pada Gambar 5.

Tabel 4. Perbandingan Performa Klasifikasi YOLOv13-Nano vs KNN Baselines

Metode / Model	Akurasi (%)	Precision (%)	Recall (%)	F1-Score (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Waktu Inferensi (%)
KNN (Full Image - Baseline 1)	9,90%	13,11%	9,91%	9,45%	N/A	N/A	43,05 ms
KNN (Hand Cropped - Baseline 2)	11,23%	12,29%	11,64%	10,61%	N/A	N/A	43,06 ms
YOLOv13-Nano	89,22%	90,63%	89,62%	89,74%	83,02%	59,27%	23,62 ms



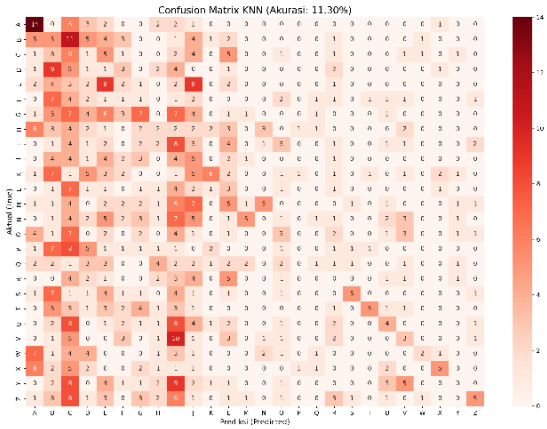
Gambar 5. Grafik Perbandingan Performa Akurasi dan F1-Score YOLOv13-Nano vs KNN Baselines

Perbandingan performa klasifikasi secara kuantitatif antara model YOLOv13-Nano dengan kedua baseline KNN disajikan pada Tabel 4. Berbeda dengan algoritma KNN yang merupakan pengklasifikasi murni tanpa keluaran koordinat kotak pembatas, arsitektur YOLOv13-Nano dievaluasi menggunakan metrik deteksi objek tambahan berupa *mean Average Precision* (mAP). Model YOLOv13-Nano mencatatkan nilai mAP@0.5 sebesar 83,02% dan mAP@0.5:0.95 sebesar 59,27% pada data validasi akhir.

Berdasarkan Tabel 4, terlihat adanya kesenjangan performa yang sangat signifikan antara metode usulan dengan baseline tradisional. Model YOLOv13-Nano mencapai akurasi klasifikasi sebesar 89,22% dan F1-Score 89,74%. Angka ini jauh mengungguli KNN (Full Image) yang hanya memperoleh akurasi 9,90% and F1-Score 9,45%.

Kesenjangan performa ini tetap bertahan bahkan ketika KNN diberikan keuntungan informasi lokalisasi tangan yang sempurna pada skenario KNN (Hand Cropped), yang hanya mencatatkan akurasi sebesar 11,23% dan F1-Score 10,61%. Kegagalan KNN dalam mengklasifikasikan abjad BISINDO disebabkan oleh keterbatasan metode berbasis jarak Euclidean dalam menangani variasi spasial yang kompleks (seperti rotasi ringan, pergeseran posisi, dan kemiripan bentuk geometris jari antar-abjad).

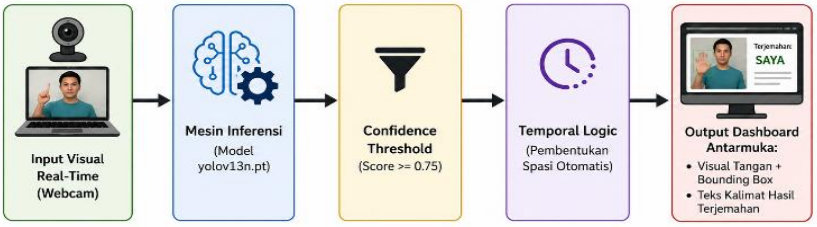
Selain itu, dari aspek kecepatan, YOLOv13-Nano memiliki waktu inferensi yang lebih cepat yaitu 23,62 ms per citra dibanding KNN (43,06 ms). Hal ini dikarenakan YOLOv13 melakukan ekstraksi fitur secara paralel melalui arsitektur CNN *feed-forward*, sedangkan KNN harus menghitung jarak Euclidean terhadap seluruh data latih saat proses query berlangsung di setiap frame.



Gambar 6. Visualisasi Confusion Matrix Klasifikasi KNN (Hand Cropped) pada Dataset BISINDO

3.3 Arsitektur Sistem *Real-Time*

Keberhasilan metrik statis di atas direalisasikan ke dalam wujud penerapan sistem *real-time* berbasis *web* menggunakan kerangka kerja asinkron FastAPI. Model YOLOv13 (yolov13n.pt) diintegrasikan sebagai mesin inferensi utama yang memfilter tebakan dengan *Confidence Threshold* ketat sebesar 0,75 untuk mengeleminasi deteksi benda mati di latar belakang. Guna menjembatani deteksi bingkai tunggal menjadi kalimat linguistik yang bermakna, sistem mengimplementasikan *Temporal Logic* (Logika Temporal). Logika algoritmik ini merekam jeda waktu (*timestamp*) tangan yang turun/hilang dari kamera selama durasi tertentu (1,0 detik) untuk secara otomatis menyisipkan karakter [SPASI]. Hasil akhir dirender ke dalam *Dashboard* bergaya *Glassmorphism* yang interaktif, seperti yang diilustrasikan pada alur Gambar 7.



Gambar 7. Arsitektur Alur Logika Pemrosesan Sistem *Real-Time*

3.4 Pengujian dan Pembahasan

Pengujian fungsionalitas penerapan sistem dilakukan secara bertahap. Pada tahap awal, pengujian difokuskan pada deteksi abjad tunggal secara *real-time*. Pada skenario ini, subjek meragakan 26 kelas alfabet BISINDO (A–Z) secara langsung di depan kamera untuk memvalidasi kecepatan respons dan ketepatan klasifikasi model dasar.

Hasil evaluasi menunjukkan performa yang sangat responsif. Keseluruhan sampel dari A hingga Z mencatatkan status prediksi Valid, di mana model secara konsisten berhasil memproyeksikan kotak pembatas (*bounding box*) tepat pada area tangan dan mengklasifikasikan label gestur secara akurat. Keandalan deteksi ini divalidasi secara matematis oleh nilai skor konfidensi (*confidence score*) yang konsisten berada di atas batas 85% untuk mayoritas kelas alfabet pada pengujian visual tunggal, sebagaimana dirangkum pada Tabel 5. Bahkan, sistem mampu menyentuh angka kepastian tinggi pada beberapa lekukan huruf yang rumit, seperti 'F' (96,7%) dan 'P' (95%), sebagaimana dirangkum pada Tabel 5.

Keberhasilan pada tahap deteksi huruf tunggal ini menjadi fondasi empiris yang sangat krusial. Skor konfidensi yang tinggi membuktikan bahwa penerapan model YOLOv13-Nano siap

secara operasional, sekaligus menjadi syarat wajib sebelum sistem dievaluasi lebih jauh menggunakan skenario deteksi dinamis yang jauh lebih kompleks pada tahapan berikutnya.

Tabel 5. Hasil Pengujian Deteksi Abjad Tunggal Secara Real-Time

Letter Input	**A**	**B**	**C**	**D**	**E**	**F**
Image						
Confidence Score (%)	92.9%	94.2%	94.6%	86%	93%	96.7%
Letter Input	**G**	**H**	**I**	**J**	**K**	**L**
Image						
Confidence Score (%)	89.8%	95%	93.6%	90.8%	93.1%	88.2%
Letter Input	**M**	**N**	**O**	**P**	**Q**	**R**
Image						
Confidence Score (%)	86.3%	91.2%	93.1%	95%	88.5%	93.9%
Letter Input	**S**	**T**	**U**	**V**	**W**	**X**
Image						
Confidence Score (%)	91.8%	90.1%	89.8%	94.9%	93%	91.2%
Letter Input	**Y**	**Z**				
Image						
Confidence Score (%)	93.3%	85.7%				

3.5 Pengujian Ketahanan terhadap Variasi Lingkungan

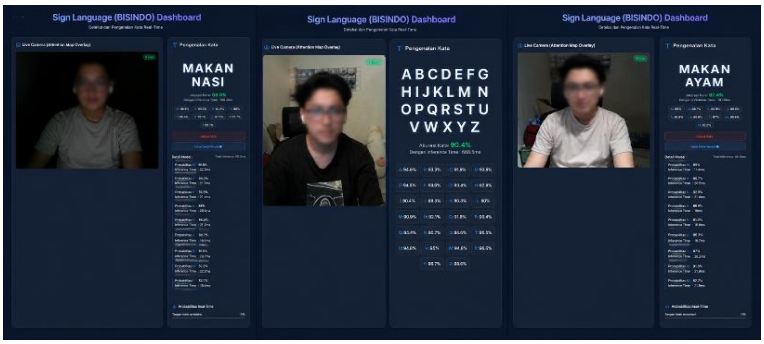
Untuk menguji ketahanan sistem secara *real-time*, evaluasi dilakukan melalui 5 kali pengulangan ejaan kata target 'MAKAN AYAM' (total 50 karakter uji) untuk masing-masing skenario lingkungan nyata. Pada kondisi normal (300–400 lux), sistem menghasilkan rata-rata Akurasi Kata (*Word Accuracy*) sebesar 87,4% dengan total kesalahan ejaan sebanyak 6 karakter dari keseluruhan percobaan. Logika temporal sistem bekerja dengan merekam jeda waktu antartangan dan menyisipkan spasi otomatis secara dinamis tanpa intervensi tombol manual.

Selanjutnya, untuk menguji kemampuan fokus arsitektur YOLOv13-Nano, sistem dihadapkan pada gangguan eksternal berupa lingkungan dengan latar belakang yang kompleks (*cluttered background*). Meskipun kamera menangkap banyak objek benda mati dengan tekstur dan warna yang mencolok di belakang subjek, model tidak terkecoh untuk memunculkan kotak deteksi palsu (*False Positive*). Model tetap kokoh mengisolasi anatomi tangan dan mendeteksi rangkaian alfabet dengan akurasi pembacaan sebesar 90,4%.

Skenario pengujian pamungkas dilakukan pada kondisi pencahayaan ekstrem sangat redup (*low-light*). Meskipun kehilangan banyak informasi warna akibat minimnya cahaya (*underexposed*), sistem tetap mampu merangkai kalimat dengan tingkat Akurasi Kata rata-rata mencapai 88,3% (6 kesalahan karakter). Visualisasi tangkapan layar dari ketiga skenario pengujian ketahanan ini dapat ditarik perbandingannya secara komparatif pada Gambar 8.

Tingkat akurasi pada kondisi redup ekstrem yang sedikit lebih tinggi dibandingkan kondisi normal ini dianalisis terjadi karena dua faktor utama. Pertama, kegelapan ekstrem secara alami mereduksi gangguan visual (*background clutter*) di sekitar subjek, karena objek-objek latar belakang tidak tertangkap oleh sensor kamera. Hal ini menyisakan siluet tangan subjek yang mendapatkan pantulan cahaya minimal dari monitor, sehingga model YOLOv13-Nano dapat

berfokus sepenuhnya pada fitur geometri bentuk tangan tanpa distraksi visual. Kedua, perbedaan akurasi sebesar 0,9% ini secara statistik berada dalam rentang margin kesalahan (*margin of error*) akibat keterbatasan jumlah subjek uji dan pengulangan sampel. Hal ini menunjukkan bahwa performa model cenderung stabil dan konsisten dalam mempertahankan ketepatan pembacaan baik pada kondisi terang maupun gelap.



Gambar 8. Pengujian Ketahanan Sistem (*Robustness*) kondisi *Low-Light* (Kiri), Latar Belakang Kompleks (Tengah) dan Logika Pembentukan Kata (Kanan)

3.6 *Discussion*

Evaluasi performa YOLOv13-Nano pada penelitian ini mencatatkan nilai rata-rata Precision sebesar 90,63%, Recall 89,62%, F1-Score 89,74%, dan waktu inferensi offline sebesar 23,62 ms per citra. Berdasarkan tinjauan literatur terhadap penelitian terdahulu, hasil evaluasi ini menawarkan alternatif performa yang kompetitif. Sebagai perbandingan tidak langsung, Farhana et al. [6] melaporkan akurasi sebesar 89,74% menggunakan YOLOv8, sedangkan Renaningtias et al. [7] yang menggunakan YOLOv7 mencatatkan metrik evaluasi offline yang tinggi namun mengidentifikasi adanya kerentanan terhadap degradasi performa real-time akibat pergerakan objek dan variasi cahaya. Dari aspek kecepatan inferensi, model YOLOv13-Nano pada penelitian ini juga menunjukkan responsivitas yang baik dibandingkan dengan penerapan YOLOv11 oleh Azahra [8] yang melaporkan waktu jeda inferensi sekitar 30–35 ms pada perangkat mereka. Perlu digarisbawahi bahwa perbandingan metrik dengan penelitian-penelitian terdahulu tersebut tidak dilakukan secara *head-to-head* pada dataset, perangkat keras, dan kondisi eksperimen yang sama, sehingga tidak dapat disimpulkan secara mutlak mengenai keunggulan satu arsitektur atas arsitektur lainnya. Namun, hasil evaluasi mandiri ini mengonfirmasi bahwa penerapan arsitektur YOLOv13-Nano pada domain BISINDO mampu menghasilkan prototipe sistem penerjemah real-time dengan kecepatan inferensi yang responsif dan ketahanan deteksi yang konsisten pada beberapa skenario lingkungan yang diuji.

Meskipun mencatatkan keberhasilan operasional yang kuat, penelitian ini memiliki beberapa batasan (*limitations*) yang perlu diakui secara jujur. Pertama, nilai F1-Score sebesar 89,74% yang diperoleh sistem tercatat lebih rendah dibandingkan metrik evaluasi offline yang hampir sempurna dari beberapa penelitian acuan [6], [7]. Hal ini sejatinya merupakan konsekuensi (*trade-off*) dari langkah mitigasi *overfitting*, di mana model pada penelitian ini telah dilatih secara masif menggunakan 5.835 citra beraugmentasi tinggi, berbeda dengan penelitian sebelumnya yang umumnya mencetak nilai 100% semata-mata karena volume dataset yang sangat minim. Kedua, hilangnya informasi kedalaman spasial 3D akibat penggunaan kamera 2D biasa menyebabkan model masih rentan mengalami misklasifikasi pada kelas gestur dengan kepadatan tumpang tindih jari yang identik, seperti abjad 'M' dan 'N' (44% *error*). Terakhir, fungsionalitas sistem saat ini masih terbatas pada penerjemahan alfabet statis (A–Z) yang dikoordinasikan dengan logika temporal pembentukan kata (*auto-spacing*), belum mencakup pengenalan kosakata bahasa isyarat dinamis yang melibatkan pergerakan temporal kompleks seperti tubuh dan ekspresi wajah.

4. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan prototipe sistem deteksi alfabet BISINDO secara *real-time* berbasis arsitektur YOLOv13-Nano yang diintegrasikan dengan antarmuka web FastAPI. Pada skenario pengujian terbatas, sistem terbukti dapat memproses tangkapan kamera web secara responsif dan menerapkan logika temporal (*auto-spacing*) untuk merangkai ejaan huruf menjadi kata dengan capaian rata-rata Akurasi Kata (*Word Accuracy*) berkisar antara 87,4% hingga 88,3%, serta akurasi pembacaan abjad mencapai 90,4% pada latar belakang yang kompleks. Pada tahap evaluasi *offline*, model YOLOv13-Nano menunjukkan performa klasifikasi yang seimbang dengan capaian Precision sebesar 90,63%, Recall 89,62%, F1-Score 89,74%, serta mAP@0.5 sebesar 83,02%. Kinerja pengujian *real-time* juga menunjukkan stabilitas deteksi yang konsisten pada skenario variasi lingkungan nyata, baik pada kondisi pencahayaan rendah ekstrem (*extreme low-light*) di bawah 15 lux maupun gangguan visual dari latar belakang kompleks (*cluttered background*). Dengan rata-rata waktu inferensi sebesar 23,62 ms per citra, model YOLOv13-Nano menawarkan responsivitas yang baik untuk pengenalan gestur dinamis berdasarkan hasil tinjauan literatur perbandingan dengan arsitektur generasi sebelumnya (seperti YOLOv7, YOLOv8, dan YOLOv11). Sebagai pembanding langsung pada dataset dan perangkat keras yang sama, arsitektur YOLOv13-Nano menunjukkan keunggulan signifikan dibandingkan metode baseline KNN yang hanya mencapai akurasi klasifikasi maksimal 11,23% pada skenario *hand cropped* dengan waktu inferensi 45,1% lebih lambat.

Meskipun demikian, penelitian ini juga mengungkap dua keterbatasan objektif. Pertama, nilai F1-Score (89,74%) yang diperoleh sistem tercatat lebih rendah dibandingkan metrik evaluasi *offline* dari beberapa penelitian acuan terdahulu [6], yang mana hal ini merupakan konsekuensi logis (*trade-off*) dari penggunaan dataset masif (5.835 citra) beraugmentasi ekstrem guna meminimalkan gejala *overfitting*. Kedua, hilangnya informasi kedalaman 3D pada tangkapan kamera 2D biasa menyebabkan model masih rentan mengalami misklasifikasi pada gestur yang memiliki kepadatan jari identik, seperti abjad 'M' dan 'N' (tingkat kesalahan mencapai 44%).

Untuk menutupi keterbatasan sistem saat ini, penelitian selanjutnya disarankan untuk berekspansi melampaui deteksi abjad statis tunggal. Pengembangan sistem yang mampu mengenali kosakata dan kalimat BISINDO dari gerakan tangan dinamis (*video sequence*) sangat direkomendasikan. Selain itu, integrasi perangkat keras berupa sensor kamera kedalaman (*depth-sensing camera*) dapat menjadi solusi empiris untuk menekan tingkat misklasifikasi pada kelas gestur yang saling bertumpuk (seperti 'M' dan 'N').

Dari sisi kebermanfaatan produk, arsitektur YOLOv13-Nano yang berbobot ringan ini berpotensi besar untuk diimplementasikan ke dalam ekosistem aplikasi *mobile* (*smartphone*) ataupun perangkat keras tertanam (*embedded system*). Langkah ini krusial agar teknologi penerjemah cerdas ini dapat diakses secara portabel dan memberikan dampak inklusif yang nyata bagi komunitas Tuli di kehidupan sehari-hari.

DAFTAR PUSTAKA

[1] Meysa Malika and Berlianti Berlianti, "Penggunaan Bahasa Isyarat SIBI dan BISINDO untuk Siswa Tuli di Sekolah Luar Biasa Negeri Pembina Medan," *Concept: Journal of Social Humanities and Education*, vol. 4, no. 3, pp. 78–87, 2025, doi: 10.55606/concept.v4i3.2178.

[2] Badan Pusat Statistik, "Potret Penyandang Disabilitas di Indonesia: Hasil Long Form SP2020 - Badan Pusat Statistik Indonesia." Accessed: Jan. 10, 2026. [Online]. Available: https://www.bps.go.id/id/publication/2024/12/20/43880dc0f8be5ab92199f8b9/potret-penyandang-disabilitas-di-indonesia--hasil-long-form-sp2020.html

[3] A. F. Muhammad and D. Andamisari, "Strategi Komunikasi Komunitas Gerakan Kesejahteraan Tunarungu Indonesia Dalam Menyosialisasikan Bahasa Isyarat Indonesia di Jakarta," *LUGAS Jurnal Komunikasi*, vol. 8, no. 2, pp. 200–209, 2024, doi: 10.31334/lugas.v8i2.4727.

[4] M. Alaftekin, I. Pacal, and K. Cicek, "Real-time sign language recognition based on YOLO algorithm," *Neural Comput. Appl.*, vol. 36, no. 14, pp. 7609–7624, 2024, doi: 10.1007/s00521-024-09503-6.
[5] K. S. Adi, P. I. Afu, and P. W. Bondan, "Pengembangan Sistem Perhitungan Jumlah Kendaraan Berdasarkan Jenis Kendaraan Menggunakan Algoritma YOLO Secara Realtime," *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 7, pp. 166–178, 2024, doi: 10.36080/skanika.v7i2.3201.
[6] F. Farhana, A. R. E. Najaf, and R. Permatasari, "BISINDO Sign Language Interpreter System Using YOLOv8 and CNN," *bit-Tech*, vol. 8, no. 1, pp. 598–607, 2025, doi: 10.32877/bt.v8i1.2638.
[7] N. Renaningtias, F. Putra Utama, A. Nur, and A. Sobri, "Deteksi Bahasa Isyarat Indonesia (BISINDO) Pada Video dengan YOLOv7," *JSAI: Journal Scientific and Applied Informatics*, vol. 8, no. 1, 2025, doi: 10.36085/jsai.v8i1.7067.
[8] S. Azahra Putri, "Alphabet SIBI sign language recognition using YOLOv11 for real-time gesture detection," vol. 11, no. 1, pp. 123–135, Jul. 2025, doi: 10.35335/mandiri.v14i1.408.
[9] A. Munandar, Z. Yunizar, and S. Retno, "Indonesian Sign Language (BISINDO) Alphabet Detection System Using YOLO (You Only Look Once) Algorithm," *Proceedings of MICoMS*, vol. 4, pp. 1-10, 2024, doi: 10.29103/micoms.v4i.952.
[10] S. Daniels, N. Suciati, and C. Fathichah, "Indonesian Sign Language Recognition using YOLO Method," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1077, 2021, doi: 10.1088/1757-899X/1077/1/012029.
[11] D. S. Ariansyah, "Pendeteksi Kata Dalam Bahasa Isyarat Menggunakan Algoritma YOLO VERSI 8," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, pp. 2772-2778, 2024, doi: 10.23960/jitet.v12i3.4904.
[12] Y. Gao, *et al*., "Benchmarking YOLOv8 to YOLOv13 for robust hand gesture recognition in human–robot interaction," *Sci. Rep.*, vol. 15, no. 1, p. 40043, 2025, doi: 10.1038/s41598-025-23925-9.
[13] M. Lei *et al.*, "YOLOv13: Real-Time Object Detection with Hypergraph-Enhanced Adaptive Visual Perception," *Cornell University*, Sep. 2025, doi: 10.48550/arXiv.2506.17733.
[14] S. A. Sanjaya, "BISINDO Indonesian Sign Language: Alphabet Image Data," vol. 1, 2024, doi: 10.17632/YWNJPBCZ8M.1.
[15] H. D. Rimbawa, *et al*., "Artificial intelligence-based hand gesture recognition for sign language interpretation," *Jurnal Mandiri IT*, vol. 14, no. 1, pp. 76–86, 2025, doi; 10.35335/mandiri.v14i1.395.
[16] E. L. Kelana, M. Riko, A. Prasetya, and M. Zulfadhilah, "Integrating the CNN Model with the Web for Indonesian Sign Language (BISINDO) Recognition," *Journal of Applied Informatics and Computing (JAIC)*, vol. 9, no. 3, pp. 883–896, 2025, doi: 10.30871/jaic.v9i3.9345.
[17] D. Fan, *et al*., "Continuous sign language recognition algorithm based on object detection and variable-length coding sequence," *Sci. Rep.*, vol. 14, no. 1, pp. 1-21, 2024, doi: 10.1038/s41598-024-78319-0.
[18] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," *Computers*, vol. 13, no. 12, pp. 1-37, 2024, doi: 10.3390/computers13120336.
[19] L. T. Ramos and A. D. Sappa, "A Decade of You Only Look Once (YOLO) for Object Detection: A Review," pp. 1-39, 2025, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2025.3630988.
[20] K. Riehl, M. Neunteufel, and M. Hemberg, "Hierarchical confusion matrix for classification performance evaluation," *J. R. Stat. Soc. Ser. C Appl. Stat.*, vol. 72, no. 5, pp. 1394–1412, 2023, doi: 10.1093/jrsssc/qlad057.
[21] W. Shinta Sari and C. Atika Sari, "Klasifikasi Bunga Mawar Menggunakan KNN dan Ekstraksi Fitur GLCM dan HSV," *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 5, no. 2, pp. 145–156, 2022, doi: 10.36080/skanika.v5i2.2951.