

PENERAPAN METODE YOLOv11 DALAM MENDETEKSI GESTUR TANGAN ALFABET BISINDO BERBASIS WEBSITE

Muhammad Asifaq, Dita Danianti, Dhina Puspasari Wijaya, Ahmad Subhan Yazid

Informatika, Universitas Alma Ata

Jl. Brawijaya No.99, Jadan, Tamantirto, Kasihan, Bantul, Daerah Istimewa Yogyakarta, Indonesia

asifaq00@gmail.com

ABSTRAK

Bahasa Isyarat Indonesia (BISINDO) merupakan bahasa alami yang digunakan oleh komunitas Tuli di Indonesia, namun pemanfaatan teknologi untuk mendukung pembelajaran dan komunikasi BISINDO masih terbatas. Permasalahan utama yang dihadapi adalah rendahnya ketersediaan sistem otomatis yang mampu mendeteksi dan mengenali gestur alfabet BISINDO secara akurat dan *real-time*, khususnya yang dapat diakses melalui platform berbasis web. Penelitian ini bertujuan untuk mengembangkan sistem deteksi gestur tangan alfabet BISINDO berbasis website menggunakan pendekatan *object detection* berbasis *deep learning*. Metode yang digunakan adalah arsitektur YOLOv11 dengan dataset sebanyak 2.600 citra gestur alfabet BISINDO A–Z yang telah dianotasi dan dibagi ke dalam data *training*, *validation*, dan *testing*. Evaluasi model dilakukan menggunakan metrik *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95*. Hasil pengujian menunjukkan bahwa model YOLOv11 mampu menghasilkan performa deteksi yang tinggi dan stabil pada seluruh kelas alfabet BISINDO. Model terbaik selanjutnya diintegrasikan ke dalam sistem berbasis web menggunakan FastAPI sebagai *backend* dan React.js sebagai *frontend*, serta mendukung deteksi melalui webcam dan unggahan citra. Sistem yang dikembangkan mampu melakukan deteksi gestur secara *real-time* dan dapat diakses langsung melalui browser tanpa instalasi tambahan.

Kata kunci : BISINDO, YOLOv11, Deep Learning, Object Detection, Website.

1. PENDAHULUAN

Bahasa Isyarat Indonesia (BISINDO) merupakan bahasa alami yang digunakan oleh komunitas Tuli di Indonesia sebagai sarana utama dalam berkomunikasi sehari-hari [1]. Namun, keterbatasan pemahaman masyarakat umum terhadap bahasa isyarat masih menjadi hambatan utama dalam interaksi sosial antara penyandang tunarungu dan masyarakat luas. World Health Organization (WHO) melaporkan bahwa jumlah penyandang gangguan pendengaran secara global terus meningkat dan diproyeksikan mencapai 2,5 miliar orang pada tahun 2050, sehingga diperlukan solusi komunikasi yang inklusif dan berbasis teknologi [2].

Di Indonesia, sistem bahasa isyarat yang dikenal terdiri dari Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). SIBI merupakan sistem isyarat resmi yang disusun berdasarkan struktur bahasa Indonesia dan digunakan dalam konteks pendidikan formal, sedangkan BISINDO berkembang secara alami di dalam komunitas Tuli dan lebih banyak digunakan dalam komunikasi sehari-hari karena bersifat lebih ekspresif dan kontekstual, serta lebih sesuai dengan pola komunikasi visual penyandang tunarungu [3], [4]. Meskipun BISINDO lebih dominan digunakan oleh komunitas Tuli, ketersediaan media pembelajaran serta sistem pendukung komunikasi berbasis teknologi yang secara khusus mengakomodasi BISINDO masih tergolong terbatas [5].

Perkembangan *artificial intelligence*, khususnya pada bidang *computer vision* dan *deep learning*,

membuka peluang untuk mengembangkan sistem pendeteksian gestur tangan secara otomatis. Pendekatan berbasis *object detection* dinilai sesuai karena mampu melakukan proses klasifikasi dan lokalisasi objek secara simultan serta mendukung kebutuhan deteksi secara *real-time* [6]. Beberapa penelitian sebelumnya menunjukkan bahwa arsitektur You Only Look Once (YOLO) memiliki kecepatan inferensi yang tinggi dan efisiensi komputasi yang baik dibandingkan metode deteksi objek lainnya [7].

Seiring perkembangan arsitektur YOLO hingga versi terbaru, YOLOv11 dilaporkan memiliki peningkatan performa dalam mendeteksi objek berukuran kecil serta stabilitas deteksi yang lebih baik dibandingkan versi sebelumnya [8]. Namun, penerapan YOLOv11 untuk mendeteksi gestur tangan alfabet BISINDO, khususnya dalam sistem berbasis website yang dapat diakses langsung melalui browser, masih sangat terbatas.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan sistem deteksi gestur tangan alfabet BISINDO berbasis website menggunakan arsitektur YOLOv11. Rencana penyelesaian masalah dilakukan melalui pengumpulan dan anotasi dataset gestur alfabet BISINDO, pelatihan dan evaluasi model YOLOv11 menggunakan metrik *precision*, *recall*, dan *mean Average Precision* (mAP), serta integrasi model ke dalam sistem *client-server* berbasis web agar mampu melakukan deteksi gestur secara *real-time* dan mudah diakses oleh pengguna.

2. TINJAUAN PUSTAKA

2.1. Bahasa Isyarat Indonesia (BISINDO)

Bahasa Isyarat Indonesia (BISINDO) merupakan bahasa alami yang berkembang di dalam komunitas Tuli dan digunakan sebagai sarana komunikasi utama dalam kehidupan sehari-hari. Berbeda dengan Sistem Isyarat Bahasa Indonesia (SIBI) yang bersifat formal dan mengikuti struktur bahasa Indonesia lisan, BISINDO memiliki karakteristik visual yang lebih ekspresif, fleksibel, dan kontekstual [4]. BISINDO mengandalkan kombinasi gerakan tangan, posisi tubuh, serta ekspresi wajah untuk menyampaikan makna, sehingga lebih mudah dipahami oleh pengguna aktif bahasa isyarat.

Meskipun BISINDO lebih banyak digunakan oleh komunitas Tuli, dokumentasi serta media pembelajaran berbasis teknologi untuk BISINDO masih terbatas. Hal ini menyebabkan masyarakat umum mengalami kesulitan dalam mempelajari dan memahami bahasa isyarat, sehingga diperlukan dukungan teknologi yang mampu menjembatani komunikasi antara penyandang tunarungu dan masyarakat luas [1].

2.2. Deep Learning dan Convolutional Neural Network

Deep learning merupakan bagian dari *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan untuk mempelajari representasi data secara hierarkis. Salah satu arsitektur *deep learning* yang banyak digunakan dalam pengolahan citra adalah *Convolutional Neural Network* (CNN). CNN dirancang untuk mengekstraksi fitur visual seperti tepi, bentuk, dan pola secara otomatis dari citra [9].

CNN telah banyak diterapkan dalam berbagai bidang, termasuk pengenalan objek, klasifikasi citra, dan pengenalan gestur tangan. Keunggulan CNN terletak pada kemampuannya dalam menangani variasi skala, rotasi, dan pencahayaan, sehingga cocok digunakan untuk pengenalan bahasa isyarat yang memiliki karakteristik visual yang kompleks [10].

2.3. YOLO (You Only Look Once)

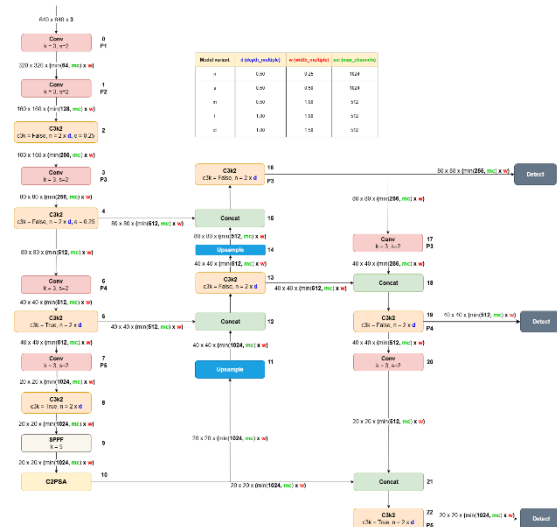
You Only Look Once (YOLO) merupakan algoritma *object detection* berbasis *deep learning* yang bekerja dengan pendekatan *single-stage detector*. Berbeda dengan metode *two-stage detector* yang memisahkan proses *region proposal* dan klasifikasi, YOLO melakukan proses deteksi objek dalam satu tahap inferensi secara end-to-end, sehingga memiliki kecepatan pemrosesan yang tinggi dan cocok untuk aplikasi real-time [11].

Pada penelitian ini digunakan arsitektur YOLOv11, yang merupakan pengembangan dari versi YOLO sebelumnya dengan peningkatan pada efisiensi komputasi dan performa deteksi. YOLOv11 dirancang untuk menghasilkan akurasi yang lebih baik, khususnya dalam mendeteksi objek berukuran kecil, tanpa mengorbankan kecepatan inferensi [8].

Karakteristik tersebut menjadikan YOLOv11 sesuai untuk diterapkan pada sistem deteksi gestur tangan alfabet Bahasa Isyarat Indonesia (BISINDO) yang membutuhkan respons cepat dan akurat secara *real-time*.

2.4. Arsitektur YOLOv11

YOLOv11 merupakan pengembangan dari algoritma *You Only Look Once* (YOLO) yang dirancang untuk meningkatkan akurasi deteksi objek dan efisiensi komputasi, khususnya pada skenario *real-time*. Secara umum, YOLOv11 tetap menggunakan struktur *backbone-neck-head*, namun dilengkapi dengan sejumlah pembaruan arsitektur dibandingkan versi YOLO sebelumnya [12].



Gambar 1. Arsitektur YOLOv11

Pada bagian *backbone*, YOLOv11 mengadopsi pendekatan *Cross Stage Partial Network* (CSP) untuk mengurangi redundansi perhitungan serta meningkatkan stabilitas aliran gradien selama proses pelatihan. Selain itu, digunakan modul C3k2 sebagai pengembangan dari modul C3 untuk menyeimbangkan kedalaman jaringan dan kecepatan inferensi. Beberapa lapisan juga menerapkan *Depthwise Separable Convolution* (DSC) guna menekan jumlah parameter dan beban komputasi tanpa mengurangi kemampuan ekstraksi fitur secara signifikan [12].

Bagian *neck* pada YOLOv11 memanfaatkan kombinasi *Path Aggregation Network* (PAN) dan *Bidirectional Feature Pyramid Network* (BiFPN) untuk melakukan penggabungan fitur multi-skala. Pendekatan ini memungkinkan informasi fitur dari berbagai tingkat resolusi digabungkan secara lebih efektif, sehingga meningkatkan kemampuan model dalam mendeteksi objek dengan ukuran yang bervariasi. Untuk memperkuat fokus pada fitur yang relevan, YOLOv11 juga mengintegrasikan modul *Cross-Channel Partial Self-Attention* (C2PSA) [12].

Pada bagian *head*, YOLOv11 melakukan prediksi kelas objek, koordinat *bounding box*, dan nilai *confidence score* secara simultan pada beberapa skala

fitur. Optimalisasi pada tahap ini memungkinkan YOLOv11 menghasilkan deteksi yang lebih stabil dengan latensi rendah [13]. Berdasarkan pembaruan arsitektur tersebut, YOLOv11 dinilai sesuai untuk diterapkan pada sistem deteksi gestur tangan alfabet Bahasa Isyarat Indonesia (BISINDO) yang membutuhkan akurasi tinggi dan respons cepat pada aplikasi berbasis website.

2.5. FastAPI

FastAPI merupakan *framework web* modern berbasis bahasa pemrograman *Python* yang dirancang untuk membangun layanan *Application Programming Interface (API)* dengan performa tinggi. *FastAPI* mendukung pemrosesan *asynchronous I/O*, validasi data otomatis berbasis *type annotation*, serta pembuatan dokumentasi API secara instan. Pada penelitian ini, *FastAPI* berperan sebagai *backend* yang menangani komunikasi antara *frontend* dan model deteksi YOLOv11, baik melalui mekanisme *request-response* berbasis *HTTP* maupun komunikasi *real-time* menggunakan *WebSocket*. Karakteristik tersebut menjadikan *FastAPI* sesuai untuk kebutuhan sistem deteksi gestur tangan BISINDO yang memerlukan respons cepat dan stabil [13].

2.6. React.js

React.js merupakan *library JavaScript* yang digunakan untuk membangun antarmuka pengguna berbasis komponen dan menerapkan konsep *single page application*. *React.js* memungkinkan pengelolaan *state* dan pembaruan tampilan secara deklaratif, sehingga antarmuka dapat diperbarui secara *real-time* tanpa perlu memuat ulang halaman. Dalam penelitian ini, *React.js* digunakan untuk membangun antarmuka sistem deteksi alfabet BISINDO, termasuk pengelolaan input webcam, unggah gambar, serta visualisasi hasil deteksi berupa *bounding box* dan label kelas. Pendekatan berbasis komponen pada *React.js* mempermudah pengembangan antarmuka yang modular, responsif, dan mudah dipelihara [14].

2.7. Penelitian Terkait

Beberapa penelitian sebelumnya telah memanfaatkan pendekatan *object detection* berbasis YOLO untuk pengenalan gestur tangan dan bahasa isyarat. Imran *et al.* mengembangkan sistem deteksi *American Sign Language* secara *real-time* menggunakan YOLOv9 dan menunjukkan bahwa pendekatan *single-stage detector* mampu memberikan kecepatan inferensi tinggi dengan akurasi yang kompetitif [15]. Selain itu, Alaftekin *et al.* menerapkan YOLOv4-CSP untuk pengenalan bahasa isyarat secara *real-time* dan menunjukkan kemampuan lokalisasi serta klasifikasi gestur secara simultan, sehingga lebih efisien dibandingkan metode klasifikasi konvensional [16].

Dalam konteks Indonesia, beberapa penelitian telah mengembangkan sistem pengenalan bahasa

isyarat SIBI dan BISINDO menggunakan CNN maupun YOLO versi sebelumnya. Namun, sebagian besar penelitian masih berfokus pada klasifikasi citra statis atau belum mengimplementasikan sistem dalam bentuk aplikasi berbasis web yang mendukung deteksi *real-time* [5], [17]. Selain itu, pemanfaatan arsitektur YOLOv11 untuk deteksi gestur tangan alfabet BISINDO masih sangat terbatas, sehingga penelitian ini diharapkan dapat mengisi celah tersebut.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental dengan tujuan mengembangkan dan menguji sistem deteksi gestur tangan alfabet Bahasa Isyarat Indonesia (BISINDO) berbasis website menggunakan metode *deep learning*. Alur penelitian meliputi tahapan pengumpulan dataset, pelabelan data, pelatihan model YOLOv11, evaluasi kinerja model, serta implementasi sistem ke dalam arsitektur *client-server* berbasis web. Tahapan penelitian disusun secara sistematis untuk memastikan model yang dihasilkan memiliki performa yang akurat dan dapat digunakan secara *real-time*.

3.1. Dataset dan Pengolahan Data

Dataset yang digunakan dalam penelitian ini berupa citra gestur tangan alfabet BISINDO A–Z yang dikumpulkan melalui proses observasi dan verifikasi data. Setiap citra merepresentasikan satu kelas alfabet BISINDO dan telah melalui proses seleksi untuk memastikan kualitas data. Selanjutnya, dataset dilakukan proses pelabelan (*annotation*) menggunakan format *bounding box* sesuai dengan kebutuhan algoritma YOLO. Dataset yang telah dilabeli kemudian dibagi menjadi data *training*, *validation*, dan *testing*.

Tabel 1. Pembagian Dataset Gestur Alfabet BISINDO

Dataset	Jumlah Data	Persentase
Training	2.080	80%
Validation	260	10%
Testing	260	10%

3.2. Pelatihan Model YOLOv11

Model deteksi objek dilatih menggunakan arsitektur YOLOv11 dengan memanfaatkan *framework* PyTorch. Proses pelatihan menerapkan pendekatan *transfer learning* dengan menggunakan bobot awal (*pretrained weights*) untuk mempercepat konvergensi dan meningkatkan performa model. Selain itu, diterapkan teknik *data augmentation* seperti *mosaic* dan *mixup* untuk meningkatkan keragaman data dan mengurangi risiko *overfitting*. Proses pelatihan dilakukan dengan pengaturan *hyperparameter* yang disesuaikan berdasarkan hasil eksperimen awal.

Tabel 2. Konfigurasi Pelatihan Model YOLOv11

Parameter	Nilai
Arsitektur Model	YOLOv11
Input Size	640 × 640
Batch size	16
Epoch maksimum	100
Optimizer	Adam
Learning rate awal	0.001
Scheduler	Cosine Annealing
Early stopping	Patience = 10

3.3. Evaluasi Kinerja Model

Evaluasi kinerja model dilakukan menggunakan metrik *Intersection over Union (IoU)*, *precision*, *recall*, *F1-score*, dan *mean Average Precision (mAP)*. Nilai *IoU* digunakan untuk mengukur tingkat kesesuaian antara *bounding box* hasil prediksi dan *ground truth*, yang dirumuskan sebagai berikut:

$$IoU = \frac{Area(B_{pred} \cap B_{gt})}{Area(B_{pred} \cup B_{gt})} \quad (1)$$

Nilai *precision* dan *recall* digunakan untuk mengukur ketepatan dan kelengkapan hasil deteksi, yang dirumuskan sebagai:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Selanjutnya, *F1-score* digunakan untuk menilai keseimbangan antara *precision* dan *recall*:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Performa keseluruhan model diukur menggunakan *mean Average Precision (mAP)*, yang dihitung sebagai rata-rata nilai *Average Precision (AP)* dari seluruh kelas alfabet BISINDO:

$$AP = \int_0^1 Precision(Recall) d(Recall) \quad (5)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (6)$$

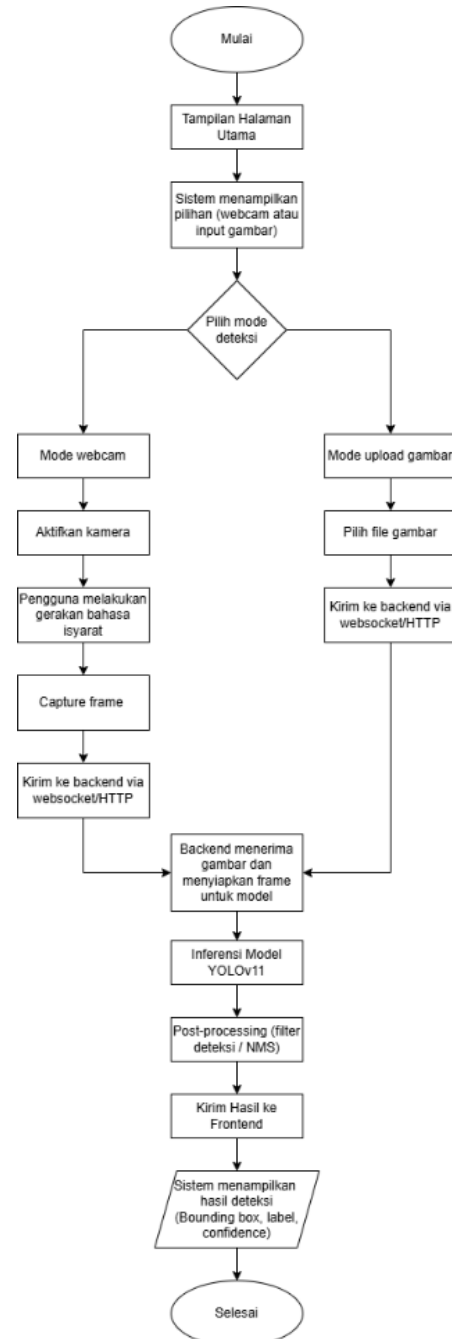
Evaluasi dilakukan pada data *validation* dan *testing* untuk menentukan konfigurasi model terbaik yang selanjutnya diimplementasikan ke dalam sistem website.

3.4. Implementasi Sistem Website

Sistem deteksi gestur BISINDO diimplementasikan menggunakan arsitektur *client-server*. Pada sisi *backend*, digunakan *FastAPI* untuk menangani proses inferensi model YOLOv11 serta komunikasi data melalui *HTTP* dan *WebSocket*. Sementara itu, sisi *frontend* dikembangkan

menggunakan *React.js* untuk membangun antarmuka pengguna yang responsif dan mendukung pembaruan tampilan secara *real-time*. Sistem mendukung dua mode deteksi, yaitu deteksi melalui webcam dan deteksi menggunakan citra statis yang diunggah oleh pengguna.

3.5. Alur Sistem Deteksi



Gambar 2. Flowchart umum sistem

Alur kerja sistem deteksi gestur tangan BISINDO ditunjukkan pada Gambar 2. Sistem dimulai ketika pengguna mengakses aplikasi website melalui *browser* dan memilih mode deteksi, yaitu menggunakan webcam atau unggahan citra. Data citra yang diterima

kemudian dikirim ke sisi *backend* melalui layanan API untuk diproses oleh model YOLOv11.

Pada sisi *backend*, citra masukan diproses melalui tahapan *preprocessing* dan inferensi model untuk menghasilkan prediksi berupa kelas alfabet BISINDO, koordinat *bounding box*, dan nilai *confidence score*. Hasil deteksi selanjutnya dikirim kembali ke *frontend* dalam format data terstruktur dan ditampilkan secara *real-time* kepada pengguna. Alur ini memastikan sistem mampu melakukan deteksi gestur secara cepat dan responsif sesuai dengan kebutuhan aplikasi berbasis web.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Eksperimen Pelatihan Model

Eksperimen dilakukan menggunakan tiga skenario pelatihan untuk mengevaluasi pengaruh *transfer learning* dan *data augmentation* terhadap performa deteksi gestur tangan alfabet BISINDO. Seluruh skenario menggunakan arsitektur YOLOv11n dengan konfigurasi *hyperparameter* yang sama, sehingga perbedaan performa yang diperoleh dipengaruhi oleh strategi inisialisasi bobot dan penggunaan augmentasi data.

Tabel 3. Skenario Eksperimen Pelatihan Model

Skenario	Inisialisasi Bobot	Data Augmentasi
S1	Pretrained	Ya
S2	Pretrained	Tidak
S3	Scratch	Ya

4.2. Perbandingan Hasil Pelatihan

Hasil evaluasi terbaik dari masing-masing skenario ditunjukkan pada Tabel 4. Evaluasi dilakukan menggunakan metrik *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95* pada *validation set*.

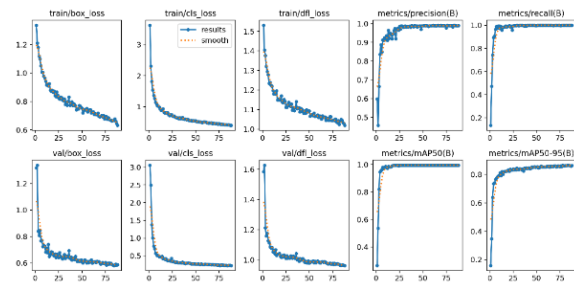
Tabel 3. Hasil Evaluasi Setiap Skenario

Skenario	Precision	Recall	mAP@0.5	mAP@0.5:0.95
S1	0,991	1,000	0,995	0,865
S2	0,990	0,996	0,995	0,854
S3	0,992	0,999	0,995	0,858

Berdasarkan Tabel 4, skenario S1 (*pretrained* dengan *data augmentation*) menghasilkan performa terbaik pada seluruh metrik evaluasi. Nilai *mAP@0.5* sebesar 0,995 menunjukkan kemampuan deteksi yang sangat tinggi, sementara *mAP@0.5:0.95* sebesar 0,865 mengindikasikan kualitas lokalisasi *bounding box* yang lebih baik dibandingkan skenario lainnya.

4.3. Analisis Kurva Pelatihan

Perkembangan performa model selama proses pelatihan divisualisasikan melalui kurva *training loss*, *validation loss*, *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95*. Kurva pelatihan untuk skenario terbaik (S1) ditunjukkan pada Gambar 3.



Gambar 3. Kurva pelatihan skenario terbaik (S1)

Kurva *loss* menunjukkan penurunan yang stabil hingga mencapai konvergensi tanpa indikasi *overfitting*. Selain itu, kurva *mAP@0.5* dan *mAP@0.5:0.95* mengalami peningkatan signifikan pada fase awal pelatihan dan mulai stabil setelah epoch ke-70. Hal ini menunjukkan bahwa kombinasi *transfer learning* dan *data augmentation* mampu mempercepat konvergensi model serta meningkatkan kemampuan generalisasi terhadap variasi pose dan pencahayaan.

4.4. Analisis Performa Deteksi per Kelas Alfabet BISINDO

Analisis performa per kelas alfabet BISINDO dilakukan pada *test set* menggunakan metrik *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95* untuk mengevaluasi kemampuan klasifikasi dan lokalisasi model. Ringkasan metrik per kelas disajikan pada Tabel 4.

Tabel 4. Nilai Metrik per Kelas Alfabet BISINDO

Kelas	Precision	Recall	mAP@0.5	mAP@0.5:0.95
all	0.992	1.000	0.995	0.863
A	0.991	1.000	0.995	0.924
B	0.990	1.000	0.995	0.925
C	0.994	1.000	0.995	0.838
D	0.989	1.000	0.995	0.887
E	0.995	1.000	0.995	0.849
F	0.990	1.000	0.995	0.894
G	0.998	1.000	0.995	0.791
H	0.991	1.000	0.995	0.850
I	0.991	1.000	0.995	0.811
J	0.996	1.000	0.995	0.869
K	0.992	1.000	0.995	0.819
L	0.991	1.000	0.995	0.909
M	0.989	1.000	0.995	0.890
N	0.989	1.000	0.995	0.887
O	0.990	1.000	0.995	0.832
P	0.991	1.000	0.995	0.854
Q	0.990	1.000	0.995	0.863
R	0.991	1.000	0.995	0.816
S	0.995	1.000	0.995	0.882
T	0.995	1.000	0.995	0.801
U	0.992	1.000	0.995	0.863
V	0.992	1.000	0.995	0.904
W	0.989	1.000	0.995	0.922
X	0.992	1.000	0.995	0.827
Y	0.992	1.000	0.995	0.843
Z	0.993	1.000	0.995	0.881

Hasil pada Tabel 4 menunjukkan bahwa model YOLOv11 mampu mencapai performa deteksi yang tinggi dan konsisten pada seluruh kelas alfabet BISINDO. Nilai *precision*, *recall*, dan *mAP* yang diperoleh pada berbagai ambang *Intersection over Union* (IoU) menunjukkan bahwa model tidak hanya akurat dalam mengenali kelas gestur, tetapi juga memiliki kemampuan lokalisasi objek yang baik. Hal ini mencerminkan kemampuan generalisasi model yang stabil pada berbagai variasi pose tangan dan kondisi pengambilan citra.

4.5. Tampilan Antarmuka Sistem Website

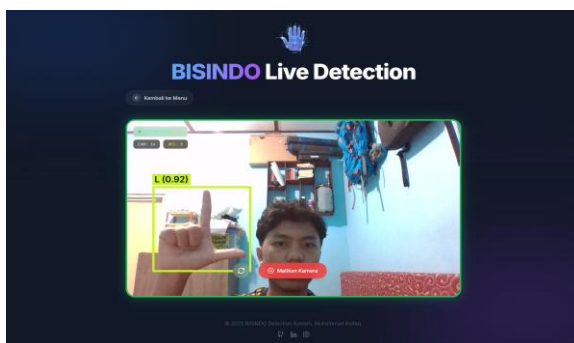
Sistem deteksi alfabet Bahasa Isyarat Indonesia (BISINDO) diimplementasikan dalam bentuk aplikasi berbasis *web* dengan antarmuka yang dirancang sederhana dan mudah digunakan. Halaman utama sistem menyediakan dua mode input, yaitu deteksi gestur secara *real-time* melalui *webcam* serta deteksi gestur melalui unggahan citra. Pada mode *real-time*, tampilan video dari kamera ditampilkan secara langsung pada halaman aplikasi, disertai hasil deteksi berupa *bounding box* dan label kelas alfabet BISINDO.



Gambar 4. Tampilan halaman utama



Gambar 5. Tampilan halaman deteksi webcam

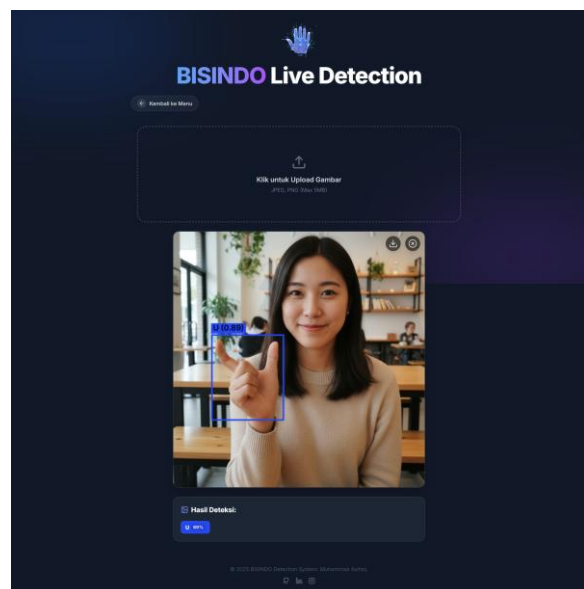


Gambar 6. Tampilan hasil deteksi webcam

Pada mode unggahan citra, pengguna dapat mengunggah gambar gestur tangan melalui antarmuka yang tersedia. Sistem kemudian memproses citra tersebut dan menampilkan hasil deteksi dengan visualisasi *bounding box* dan label kelas yang seragam dengan mode *real-time*. Seluruh proses dapat diakses langsung melalui *browser* tanpa memerlukan instalasi aplikasi tambahan, sehingga antarmuka sistem mendukung penggunaan yang fleksibel pada berbagai perangkat.



Gambar 7. Tampilan halaman deteksi *upload* gambar



Gambar 8. Tampilan hasil deteksi *upload* gambar

4.6. Pengujian Sistem

Pengujian sistem berbasis *web* dilakukan menggunakan metode *black-box testing* untuk memastikan fungsionalitas sistem berjalan sesuai dengan spesifikasi. Sistem diuji pada mode deteksi *real-time* menggunakan *webcam* serta mode unggahan citra statis.

Tabel 4. Hasil Pengujian *Blackbox-Testing*

No	Fitur yang Diuji	Skenario Pengujian	Hasil
1	Akses aplikasi	Aplikasi diakses melalui browser	Berhasil
2	Aktivasi webcam	Kamera aktif dan menampilkan stream video	Berhasil

No	Fitur yang Diuji	Skenario Pengujian	Hasil
3	Deteksi <i>real-time</i>	Gestur terdeteksi dan ditampilkan	Berhasil
4	Unggah citra	Gambar berhasil diunggah	Berhasil
5	Deteksi citra statis	Gestur pada gambar terdeteksi	Berhasil
6	Visualisasi hasil	Bounding box dan label tampil	Berhasil
7	Respons sistem	Sistem merespons tanpa error	Berhasil

Hasil pengujian menunjukkan bahwa seluruh fitur utama sistem berfungsi dengan baik. Sistem mampu menangani input video *real-time* dari *webcam* maupun input citra statis melalui unggahan gambar. Integrasi antara *frontend* berbasis *React.js* dan *backend* berbasis *FastAPI* berjalan stabil, dengan waktu respons yang memadai untuk mendukung penggunaan secara *real-time*.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengembangkan sistem deteksi gestur tangan alfabet Bahasa Isyarat Indonesia (BISINDO) secara *real-time* menggunakan arsitektur *YOLOv11* dengan performa deteksi yang tinggi dan konsisten pada seluruh kelas alfabet berdasarkan hasil pengujian *test set*. Sistem diimplementasikan dalam bentuk aplikasi berbasis *web* yang mendukung input *webcam* dan unggahan citra sehingga dapat diakses melalui *browser* tanpa instalasi tambahan. Untuk pengembangan selanjutnya, penelitian dapat diarahkan pada perluasan dataset, peningkatan variasi kondisi pengambilan citra, serta pengembangan deteksi gestur dinamis agar sistem mampu mengenali kosakata atau kalimat BISINDO secara berurutan dan lebih aplikatif dalam penggunaan nyata.

DAFTAR PUSTAKA

- [1] A. S. Nugraheni, A. P. Husain, and H. Unayah, "Optimalisasi Penggunaan Bahasa Isyarat Dengan Sibi Dan Bisindo Pada Mahasiswa Difabel Tunarungu Di Prodi Pgmi Uin Sunan Kalijaga," *J. Holistika*, vol. 5, no. 1, p. 28, 2023, doi: 10.24853/holistika.5.1.28-33.
- [2] World Health Organization, *World Report On Hearing*. 2021. [Online]. Available: <https://www.who.int/publications/i/item/world-report-on-hearing>
- [3] Kemendikbud, "Kamus Sistem Isyarat Bahasa Indonesia (SIBI)." [Online]. Available: <https://pmpk.kemdikbud.go.id/sibi/profil>
- [4] Z. E. Subandi and C. Febianca, "Tindak Komunikasi Sumber Daya Manusia Tunarungu (Studi Kasus Pada Sunyi House of Coffee and Hope)," *IKON J. Komun.*, vol. 25, no. 2, pp. 157–175, 2020, [Online]. Available: <https://journals.upi-yai.ac.id/index.php/IKON/article/view/1804/1487>
- [5] D. A. Virgiawan, "PENGENALAN BAHASA ISYARAT MENGGUNAKAN DETEKSI OBJEK DEEP LEARNING," 2024.
- [6] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years : A Survey," pp. 1–22, 2023.
- [7] C. Wang and H. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020.
- [8] M. A. R. Alif, "YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems," 2024, [Online]. Available: <http://arxiv.org/abs/2410.22898>
- [9] Y. Bengio, Y. Lecun, and G. Hinton, "Deep learning for AI," *Commun. ACM*, vol. 64, no. 7, pp. 58–65, 2021, doi: 10.1145/3448250.
- [10] W. Khan and D. L. K. Vishwamitra, "Image Classification using modified Convolutional Neural Network," vol. 3, pp. 3465–3472, 2024.
- [11] R. Sapkota, M. F. Rizwan, and Q. Chetan, "YOLO advances to its genesis : a decadal and comprehensive review of the You Only Look Once (YOLO) series," 2025.
- [12] R. Khanam and M. Hussain, "Yolov11: AN OVERVIEW OF THE KEY ARCHITECTURAL ENHANCEMENTS," vol. 2024, pp. 1–9, 2024.
- [13] Ultralytics, "YOLO11 - Ultralytics YOLO Docs," Ultralytics Documentation. [Online]. Available: <https://docs.ultralytics.com/models/yolo11/>
- [14] React Team, "React – A JavaScript library for building user interfaces." [Online]. Available: <https://react.dev>
- [15] M. Shashishekhara, H. Hamza, and I. Etit-kit, "Real Time American Sign Language Detection Using Yolo-v9" 2024.
- [16] 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.
- [17] I. Amri, "Implementasi Algoritma Convolutional Neural Network Untuk Menerjemahkan Bahasa Isyarat," *J. Multidisiplin Saintek Vol.*, vol. 2, no. 9, pp. 70–87, 2024, [Online]. Available: <https://ejournal.warunayama.org/kohesi>