

Optimasi Arsitektur DETR dan Augmentasi On-the-Fly untuk Deteksi Alfabet BISINDO

Mukhamad Aziz Firmansyah¹, Fetty Tri Anggraeny², Yisti Vita Via³

^{1,2,3}Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia
¹22081010030@student.upnjatim.ac.id, ²fettyanggraeny.if@upnjatim.ac.id, ³yistivita.if@upnjatim.ac.id

Abstrak

Penelitian ini membahas penerapan Detection Transformer (DETR) yang dioptimasi dan dikombinasikan dengan augmentasi data *on-the-fly* untuk mendeteksi alfabet Bahasa Isyarat Indonesia (BISINDO) pada skenario *real-time*. Eksperimen dilakukan menggunakan dataset alfabet BISINDO yang terdiri dari 1.300 data dari 26 kelas (A–Z) dengan anotasi *bounding box* yang dikumpulkan secara mandiri pada lingkungan *indoor* terkontrol. Optimasi DETR dilakukan dengan menyederhanakan arsitektur dari konfigurasi standar menjadi 1 *layer transformer encoder*, 1 *layer transformer decoder*, dan 25 *object queries* untuk meningkatkan efisiensi inferensi. Augmentasi data *on-the-fly* diterapkan pada tahap pelatihan untuk memperkaya variasi data. Evaluasi dilakukan menggunakan metrik berbasis COCO menggunakan pustaka PyCOCOtools pada data uji serta pengujian *real-time* menggunakan input video *frame-per-frame* dari kamera. Hasil pengujian menunjukkan bahwa model tanpa augmentasi memperoleh nilai AP50 sebesar 0,987 pada data uji, sedangkan model dengan augmentasi *on-the-fly* memperoleh nilai AP50 sebesar 0,848. Meskipun nilai AP50 pada evaluasi statis menurun, hasil pengujian *real-time* menunjukkan bahwa model dengan augmentasi data *on-the-fly* menghasilkan deteksi yang lebih sesuai dibandingkan model tanpa augmentasi. Selain itu, optimasi arsitektur menghasilkan kecepatan inferensi hingga 58 FPS pada GPU dan 8 FPS pada CPU. Hasil ini menunjukkan bahwa penerapan DETR yang dioptimasi dan augmentasi data *on-the-fly* dapat digunakan untuk mendeteksi alfabet BISINDO pada skenario *real-time*.

Kata kunci: DETR, augmentasi data on-the-fly, BISINDO, *real-time*

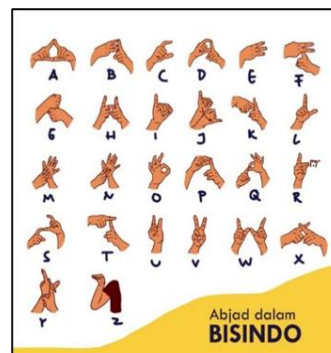
1. Pendahuluan

Bahasa merupakan sarana utama manusia dalam membangun komunikasi dan interaksi sosial (Maghfiroh, 2022). Namun, tidak semua orang dapat berkomunikasi secara lisan (Ulfah, 2023). Penyandang disabilitas pendengaran memiliki keterbatasan dalam komunikasi verbal sehingga membutuhkan media komunikasi alternatif berbasis visual (Agustin et al., 2023). Kondisi ini mendorong penggunaan bahasa isyarat sebagai sistem komunikasi utama dalam komunitas tuli (Khotijah et al., 2023).

Bahasa isyarat memanfaatkan gerakan tangan, ekspresi wajah, serta postur tubuh untuk merepresentasikan makna secara visual (Sari et al., 2023). Salah satu komponen dasar dalam bahasa isyarat adalah alfabet atau *fingerspelling*, yang merepresentasikan alfabet secara individual melalui pose tangan statis. *Fingerspelling* banyak digunakan untuk mengeja nama, istilah teknis, atau kata yang belum memiliki gestur baku, sehingga memiliki karakteristik visual yang relatif konsisten dan sesuai untuk pendekatan pengenalan berbasis citra (Agata et al., 2024).

Di Indonesia, terdapat dua sistem bahasa isyarat yang dikenal, yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO) (Subkhi et al., 2024). SIBI digunakan secara formal dalam konteks pendidikan, sedangkan BISINDO berkembang secara alami dan lebih luas

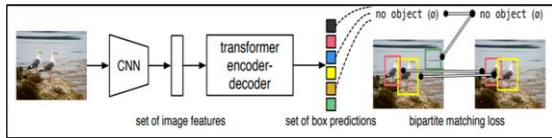
digunakan dalam komunikasi sehari-hari komunitas tuli (Trisianto & Limantara, 2024). Secara linguistik, bahasa isyarat memiliki struktur yang berbeda dari bahasa lisan karena menggunakan modalitas visual sebagai media utama penyampaian informasi (Arisandi & Satya, 2022). Pada penelitian ini, BISINDO dipilih sebagai fokus karena merepresentasikan sistem bahasa isyarat yang paling umum digunakan di komunitas tuli. Setiap alfabet BISINDO memiliki pose tangan yang unik dan dapat dibedakan secara visual, sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Alfabet BISINDO (Agata et al., 2024)

Perkembangan teknologi *computer vision* membuka peluang eksperimen dalam mendeteksi alfabet bahasa isyarat. Salah satu pendekatan yang banyak digunakan untuk mendeteksi objek adalah Detection Transformer (DETR), yang mengadopsi

arsitektur transformer untuk melakukan deteksi objek secara *end-to-end* tanpa memerlukan *anchor box* maupun *non-maximum suppression* (NMS) (Carion et al., 2020). Sebagaimana alur dari DETR ditunjukkan pada Gambar 2.



Gambar 2. Alur DETR (Carion et al., 2020)

Berbagai penelitian menunjukkan bahwa DETR mampu mencapai performa yang kompetitif pada beragam skenario deteksi objek, seperti deteksi logo kendaraan (Ubaidillah et al., 2025), deteksi kWh meter analog yang terintegrasi OCR (Fitriani et al., 2025), serta deteksi konten vulgar pada citra anime (Suciati et al., 2025). Namun demikian, hingga saat ini masih terbatas kajian yang menginvestigasi penerapan dan optimasi arsitektur DETR untuk deteksi alfabet Bahasa Isyarat Indonesia (BISINDO), khususnya pada skenario penggunaan *real-time*.

Tantangan utama dalam pengembangan sistem deteksi alfabet Bahasa Isyarat Indonesia (BISINDO) terletak pada keterbatasan ketersediaan data. Pada kondisi keterbatasan data, strategi augmentasi data menjadi faktor penting untuk meningkatkan kemampuan generalisasi model (Via et al., 2023). Augmentasi data *on-the-fly* merupakan pendekatan yang melakukan transformasi citra secara dinamis selama proses pelatihan tanpa menyimpan hasil augmentasi secara fisik (Al-Fahrezi, 2025; Ricky Putra Sardika & Widhiarso, 2025). Pendekatan ini terbukti efektif dalam mempertahankan performa model pada skenario data terbatas, sebagaimana ditunjukkan pada penelitian (Fernanda, B. A., & Bastian, 2025) menunjukkan bahwa metode augmentasi data *on-the-fly* pada model YOLOv8 mampu menghasilkan performa kompetitif dengan metode augmentasi konvensional.

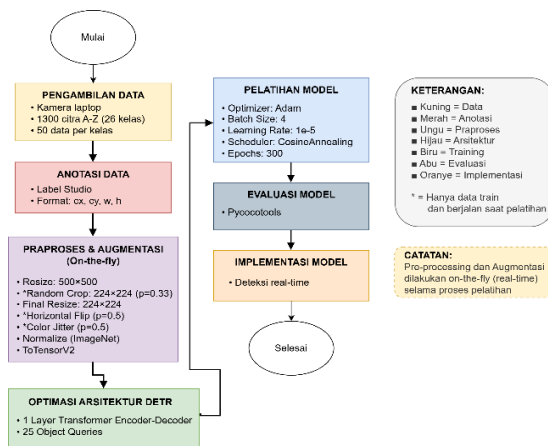
Selain aspek akurasi, evaluasi sistem deteksi bahasa isyarat juga perlu mempertimbangkan performa pada kondisi penggunaan langsung. Implementasi sistem secara *real-time* menjadikan evaluasi stabilitas deteksi pada input *frame-per-frame* dari kamera, yang lebih merepresentasikan kondisi penggunaan nyata. DETR standar dikenal memiliki kompleksitas komputasi yang tinggi karena menggunakan *multiple layers* pada *encoder* dan *decoder* (Carion et al., 2020; Yu et al., 2025). Namun, untuk konteks deteksi alfabet BISINDO yang memiliki karakteristik objek relatif sederhana dengan latar belakang terkontrol, optimasi arsitektur melalui pengurangan jumlah *layer* berpotensi meningkatkan efisiensi inferensi tanpa mengorbankan akurasi deteksi secara signifikan.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan menganalisis penerapan DETR yang dioptimasi dan dikombinasikan dengan teknik

augmentasi data *on-the-fly* untuk deteksi alfabet Bahasa Isyarat Indonesia (BISINDO), serta mengevaluasi kelayakan implementasinya pada kondisi *real-time*. Kontribusi penelitian ini meliputi: (1) optimasi arsitektur DETR menjadi konfigurasi ringan untuk meningkatkan efisiensi komputasi, (2) penerapan strategi augmentasi data *on-the-fly* untuk mengatasi keterbatasan dataset, dan (3) evaluasi stabilitas serta kecepatan inferensi pada skenario deteksi *real-time* berbasis input kamera.

2. Metode

Penelitian ini mengimplementasikan DETR yang dioptimasi dan dikombinasikan dengan strategi augmentasi data *on-the-fly* untuk mendeteksi alfabet BISINDO. Metodologi penelitian meliputi pengumpulan dan anotasi data, prapemrosesan dan augmentasi data, optimasi arsitektur DETR, pelatihan dan evaluasi model, serta implementasi deteksi pada skenario *real-time*. Alur penelitian secara umum ditunjukkan pada Gambar 3.



Gambar 3. Alur Penelitian

Gambar 3 menunjukkan alur penelitian yang diawali dari proses pengambilan data berupa citra alfabet BISINDO menggunakan kamera laptop dengan total 1.300 citra yang terdiri dari 26 kelas alfabet. Data yang telah dikumpulkan kemudian dilakukan proses anotasi menggunakan Label Studio dengan format bounding box (cx, cy, w, h).

Selanjutnya dilakukan tahap prapemrosesan dan augmentasi data secara *on-the-fly* selama proses pelatihan. Tahapan ini mencakup resize citra, random resized crop, horizontal flip, color jitter, normalisasi menggunakan standar ImageNet, serta konversi data menjadi tensor.

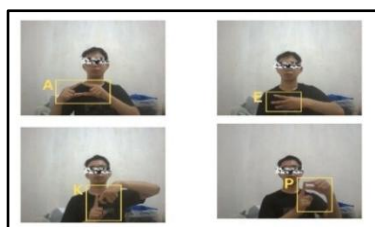
Data yang telah diproses kemudian digunakan untuk pelatihan model DETR yang telah dioptimasi dengan penyederhanaan arsitektur, yaitu penggunaan satu layer transformer encoder-decoder serta 25 object queries untuk meningkatkan efisiensi komputasi.

Setelah proses pelatihan selesai, model dievaluasi menggunakan Pycocotools untuk

mengukur performa deteksi. Tahap terakhir adalah implementasi model untuk deteksi alfabet BISINDO secara real-time.

2.1 Dataset

Dataset yang digunakan berupa citra alfabet Bahasa Isyarat Indonesia (BISINDO) yang dikumpulkan secara mandiri menggunakan kamera laptop pada lingkungan *indoor* terkontrol. Setiap citra merepresentasikan satu pose tangan statis dari 26 kelas alfabet (A–Z). Alfabet J dan R yang secara natural bersifat dinamis direpresentasikan sebagai pose statis tunggal untuk menyesuaikan dengan skema input berbasis citra tunggal (*single-frame input*). Contoh citra dataset ditunjukkan pada Gambar 4.



Gambar 4. Contoh Data BISINDO Alfabet A, E, K, dan P

Total dataset berjumlah 1.300 citra yang dibagi menjadi data latih (75%), validasi (15%), dan uji (10%). Seluruh citra disimpan dalam format JPG dengan ukuran 640×480 piksel dan dianotasi secara manual menggunakan Label Studio dan menghasilkan *bounding box* dengan koordinat ternormalisasi. Proses anotasi *bounding box* ditunjukkan pada Gambar 5.



Gambar 5. Proses Anotasi Bounding Box

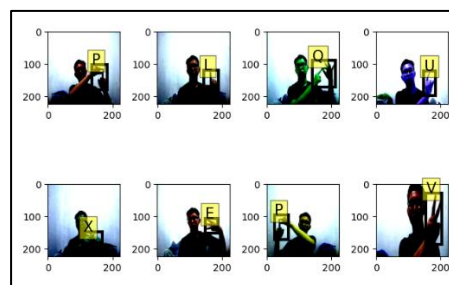
2.2 Prapemrosesan dan Augmentasi *On-the-fly*

Prapemrosesan data dilakukan untuk menyesuaikan format input citra dengan kebutuhan *backbone* ResNet-50. Seluruh citra diubah ke resolusi 224×224 piksel dan dinormalisasi menggunakan statistik ImageNet (mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]), kemudian dikonversi menjadi tensor PyTorch dengan format (C × H × W).

Untuk mengatasi keterbatasan jumlah data dan homogenitas latar belakang, penelitian ini menerapkan strategi augmentasi data *on-the-fly* pada data latih menggunakan pustaka Albumentations. Augmentasi dilakukan secara dinamis selama proses pelatihan tanpa menyimpan citra hasil transformasi,

sehingga setiap citra dapat mengalami variasi transformasi yang berbeda pada setiap *epoch*.

Transformasi yang diterapkan meliputi *random crop*, *horizontal flip*, dan *color jitter* untuk variasi orientasi tangan dan kondisi pencahayaan agar model tidak menghafal data latih. Data validasi dan uji hanya melalui tahapan prapemrosesan deterministik tanpa augmentasi acak agar hasil evaluasi mencerminkan kondisi inferensi sebenarnya dan dapat direproduksi. Visualisasi hasil prapemrosesan dan augmentasi ditampilkan pada Gambar 6.



Gambar 6. Visualisasi Data Praproses dan Augmentasi

2.3 Optimasi Arsitektur DETR

Optimasi arsitektur pada penelitian ini secara eksplisit dilakukan melalui pengurangan jumlah *layer encoder-decoder transformer* dan *object queries* dibandingkan konfigurasi standar DETR. Model diimplementasikan menggunakan satu *layer transformer encoder*, satu *layer transformer decoder*, dan 25 *object queries*, berbeda dari DETR standar yang menggunakan enam *layer encoder-decoder* dan 100 *object queries*.

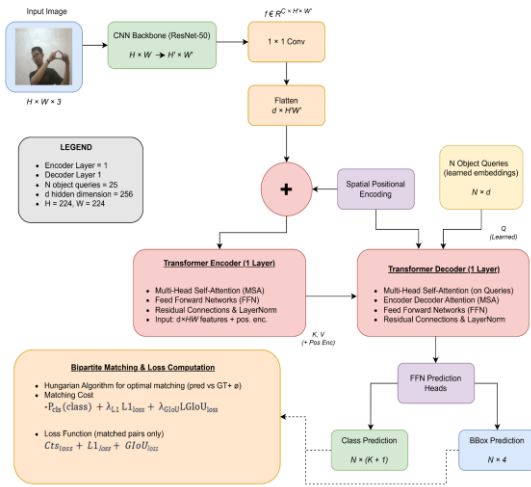
Pengurangan kompleksitas arsitektur ini dilakukan untuk menyesuaikan karakteristik citra alfabet BISINDO yang relatif sederhana dan hanya memuat satu objek tangan utama pada setiap citra. Dengan konfigurasi yang lebih ringan, diharapkan efisiensi komputasi dan kecepatan inferensi dapat meningkat tanpa mengorbankan performa deteksi secara signifikan.

Output dari *decoder* diproses oleh dua *prediction heads*, yaitu *class prediction head* untuk menghasilkan probabilitas kelas yang mencakup 26 alfabet dan satu kelas *no-object*, serta *bounding box prediction head* untuk menghasilkan koordinat *bounding box*. Proses pencocokan antara prediksi dan *ground truth* dilakukan menggunakan Hungarian Algorithm, dengan fungsi *loss* yang merupakan kombinasi dari *classification loss* dan *bounding box loss* (L1 *loss* dan GIoU *loss*). Total *loss* DETR dirumuskan pada Persamaan 1. Arsitektur DETR yang dioptimasi ditunjukkan pada Gambar 6.

$$\mathcal{L} = \sum_{i=1}^N [\mathcal{L}_{cls}(c_i, \hat{c}_{\sigma(i)}) + 1_{\{c_i \neq \emptyset\}} \mathcal{L}_{box}(b_i, \hat{b}_{\sigma(i)})] \quad (1)$$

di mana σ merupakan hasil pencocokan optimal menggunakan Hungarian Algorithm, \mathcal{L}_{cls} adalah *loss*

klasifikasi, dan \mathcal{L}_{box} merupakan kombinasi L1 loss dan GIoU loss untuk regresi *bounding box*.



Gambar 7. Arsitektur DETR (diadaptasi dari (Yu et al., 2025))

2.4 Konfigurasi Pelatihan

Model dilatih menggunakan GPU NVIDIA GeForce RTX 3080 Ti dengan optimizer Adam dan *learning rate* sebesar 1×10^{-5} . Strategi penjadwalan *learning rate* menggunakan *Cosine Annealing Warm Restarts* (CAWR) dengan parameter awal $T_0 = 30$ epoch dan $T_mult = 2$. Proses pelatihan dilakukan selama 300 *epoch* dengan *batch size* 4.

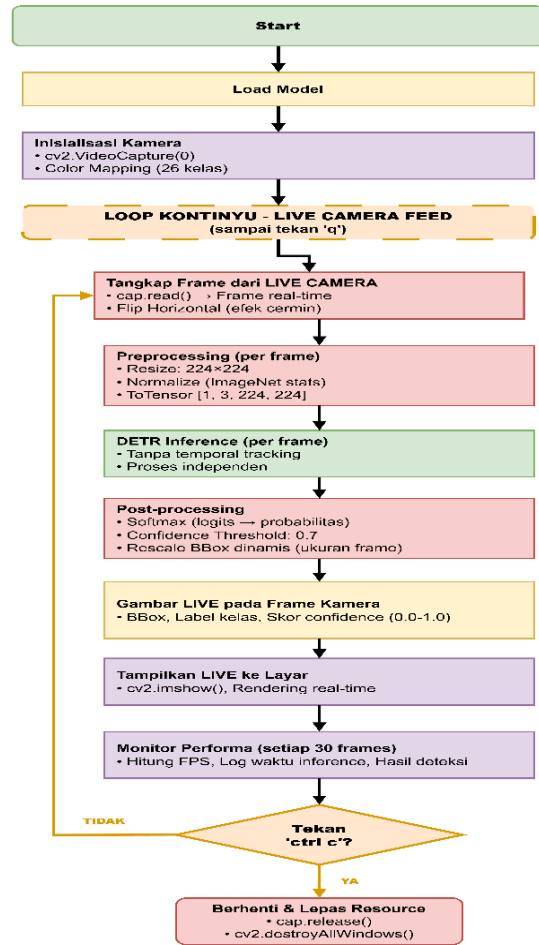
2.5 Evaluasi Model

Evaluasi performa model dilakukan menggunakan metrik standar deteksi objek yang dihitung menggunakan pustaka PyCOCOtools. Untuk mengevaluasi kelayakan implementasi pada kondisi *real-time*, digunakan metrik *Frames Per Second* (FPS) dan *latency* inferensi.

2.6 Implementasi Deteksi Real-time

Implementasi deteksi *real-time* dilakukan untuk menguji stabilitas dan kecepatan inferensi model pada kondisi penggunaan langsung. Model hasil evaluasi diimplementasikan menggunakan input video *frame-per-frame* dari kamera laptop. Setiap *frame* diproses secara independen melalui *pipeline* prapemrosesan yang sama dengan data uji, kemudian dilakukan inferensi oleh model DETR.

Hasil prediksi berupa kelas dan *bounding box* disaring menggunakan *confidence threshold* sebesar 0.7 untuk menyeimbangkan sensitivitas deteksi dan akurasi pada kondisi *real-time*. Hasil deteksi divisualisasikan secara langsung pada *live camera feed*, dan nilai FPS dicatat setiap 30 *frame* untuk memantau performa sistem. Alur implementasi deteksi *real-time* ditunjukkan pada Gambar 8.



Gambar 8. Alur Implementasi Deteksi Real-Time

2.7 Skenario Percobaan

Untuk mengevaluasi kontribusi augmentasi data *on-the-fly*, penelitian ini menerapkan dua skenario percobaan dengan konfigurasi arsitektur dan *hyperparameter* yang identik. Skenario pertama merupakan *baseline* DETR tanpa augmentasi data, sedangkan skenario kedua menggunakan augmentasi data *on-the-fly* pada data latih. Seluruh evaluasi dilakukan pada data uji yang sama untuk memastikan perbandingan yang adil.

3. Hasil dan Pembahasan

3.1 Hasil Evaluasi pada Data Uji

Evaluasi performa deteksi alfabet BISINDO dilakukan pada data uji. Pengujian dilakukan pada dua skenario, yaitu model *baseline* tanpa augmentasi data (S1) dan model dengan augmentasi data *on-the-fly* (S2).

Tabel 1. Hasil Evaluasi pada Data Uji

Skenario	AP50	AP75	AP@[0.5:0.95]
S1	0,987	0,754	0,692
S2	0,848	0,591	0,536

Berdasarkan Tabel 1, hasil model tanpa augmentasi (S1) memperoleh nilai AP50 yang lebih tinggi dibandingkan hasil model dengan augmentasi *on-the-fly* (S2). Hal ini menunjukkan bahwa model S1 sangat sesuai dengan distribusi data uji yang memiliki karakteristik serupa dengan data latih. Namun, pada metrik AP@[0.5:0.95] dan AP75, perbedaan performa antar skenario tidak bersifat proporsional terhadap nilai AP50, yang mengindikasikan adanya perbedaan karakteristik generalisasi model.

3.2 Analisis Dampak Augmentasi *On-the-fly*

Penerapan augmentasi data *on-the-fly* pada skenario S2 menghasilkan penurunan nilai AP50 dibandingkan skenario tanpa augmentasi. Fenomena ini menunjukkan bahwa model yang dilatih tanpa augmentasi cenderung menyesuaikan diri secara optimal terhadap distribusi data uji yang memiliki karakteristik visual serupa dengan data latih, khususnya pada dataset dengan latar belakang dan kondisi pencahayaan yang relatif homogen.

Namun, augmentasi *on-the-fly* memperkenalkan variasi spasial dan fotometrik secara dinamis selama proses pelatihan, sehingga memperluas distribusi data yang dipelajari model. Variasi ini mendorong model untuk mempelajari representasi visual yang lebih umum dan tidak bergantung pada karakteristik spesifik dataset pelatihan. Akibatnya, performa pada evaluasi pada data uji dapat mengalami penurunan, sementara kemampuan generalisasi pada kondisi penggunaan nyata meningkat.

Temuan ini sejalan dengan penelitian (Fernanda, B. A., & Bastian, 2025) yang mengkaji pengaruh augmentasi data *on-the-fly* pada model YOLOv8 untuk deteksi objek ritel, di mana augmentasi *on-the-fly* menghasilkan nilai mAP yang lebih rendah dibandingkan skenario tanpa augmentasi, namun mampu meningkatkan presisi model. Hasil tersebut menunjukkan bahwa augmentasi *on-the-fly* cenderung meningkatkan keyakinan model terhadap prediksi yang dihasilkan, meskipun dengan konsekuensi penurunan sensitivitas deteksi pada skenario evaluasi tertentu.

Konsistensi pola hasil ini menunjukkan bahwa dampak augmentasi data *on-the-fly* tidak terbatas pada arsitektur berbasis *convolutional neural network* seperti YOLO, tetapi juga terjadi pada arsitektur berbasis *transformer* seperti DETR. Dengan demikian, augmentasi *on-the-fly* dapat dipandang sebagai strategi yang efektif untuk meningkatkan ketahanan model pada skenario penggunaan nyata, khususnya ketika jumlah data yang terbatas dan implementasi *real-time* menjadi pertimbangan utama.

3.3 Evaluasi Implementasi Deteksi *Real-time*

Evaluasi implementasi *real-time* dilakukan untuk menilai stabilitas dan kelayakan penggunaan

model pada kondisi penggunaan langsung. Model diuji menggunakan input video *frame-per-frame* dari kamera laptop dengan *confidence threshold* sebesar 0.7. Pengujian dilakukan pada dua konfigurasi perangkat, yaitu CPU dan GPU, untuk mengukur kecepatan inferensi serta konsistensi performa sistem.

Tabel 2. Performa Real-time pada CPU dan GPU

Metriik	CPU	GPU	Peningkatan
FPS rata-rata	8,11	58,0	7,15×
Waktu inferensi rata-rata	119,2 ms	17,2 ms	6,93×
Waktu inferensi minimum	96,0 ms	14 ms	6,86×
Waktu inferensi maksimum	143,9 ms	21 ms	6,85×
Jumlah Frame	947	955	-
Jumlah Durasi	116,83 s	117,2 s	-

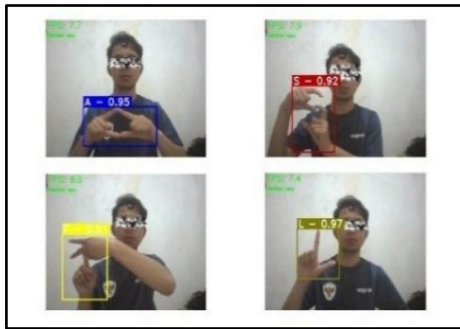
Berdasarkan Tabel 2, model dengan arsitektur DETR yang dioptimasi mampu mencapai rata-rata 58 FPS pada GPU dengan waktu inferensi rata-rata sebesar 17,2 ms per *frame*. Pada pengujian menggunakan CPU, sistem menghasilkan rata-rata 8,11 FPS dengan waktu inferensi rata-rata 119,2 ms per *frame*. Perbedaan ini menunjukkan bahwa pemanfaatan GPU memberikan peningkatan kecepatan inferensi lebih dari enam kali lipat dibandingkan CPU.

Rentang nilai *minimum* dan *maximum inference time* pada GPU relatif sempit dibandingkan CPU, yang mengindikasikan stabilitas inferensi yang lebih baik pada perangkat GPU. Hal ini penting dalam konteks aplikasi *real-time*, karena fluktuasi waktu inferensi yang besar dapat menyebabkan keterlambatan visual (*lag*) dan menurunkan pengalaman pengguna. Dengan demikian, hasil ini menunjukkan bahwa optimasi arsitektur DETR melalui pengurangan jumlah *transformer layer* dan *object queries* berkontribusi signifikan terhadap peningkatan efisiensi inferensi dan stabilitas sistem.

Pada pengujian *real-time*, perbedaan karakteristik performa juga terlihat antara dua skenario pelatihan. Model tanpa augmentasi data (S1) menunjukkan ketidakstabilan deteksi, di mana prediksi objek tidak muncul secara konsisten pada *confidence threshold* yang ditetapkan meskipun performa statis pada data uji tergolong tinggi. Sebaliknya, model dengan augmentasi data *on-the-fly* (S2) mampu menghasilkan deteksi yang lebih stabil dan konsisten pada penggunaan *real-time*.



Gambar 9. Hasil Deteksi Real-Time Model Skenario 1

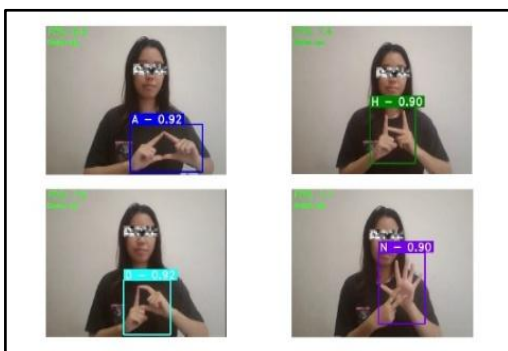


Gambar 10. Hasil Deteksi Real-Time Model Skenario 2

3.4 Uji Generalisasi pada Subjek dan Kondisi Pencahayaan Berbeda

Untuk mengevaluasi kemampuan generalisasi model lebih lanjut, pengujian *real-time* dilakukan pada subjek lain yang tidak termasuk dalam data latih, serta pada kondisi pencahayaan yang berbeda menggunakan lampu dengan variasi warna. Hasil pengujian menunjukkan bahwa model dengan augmentasi data *on-the-fly* tetap mampu mendeteksi alfabet BISINDO secara konsisten meskipun terdapat perbedaan warna kulit dan perubahan *color cast* akibat pencahayaan buatan.

Hasil ini menunjukkan bahwa augmentasi data *on-the-fly* berkontribusi dalam mengurangi ketergantungan model terhadap karakteristik visual spesifik pada data latih, serta meningkatkan ketahanan model terhadap variasi pencahayaan dan subjek. Temuan ini mendukung hasil pada Subbab 3.2, bahwa augmentasi data berperan penting dalam meningkatkan kelayakan sistem pada skenario penggunaan nyata.



Gambar 11. Hasil Real-Time pada Subjek yang Berbeda



Gambar 12. Hasil Real-Time pada Kondisi Pencahayaan Berbeda

3.5 Diskusi Kelayakan Optimasi DETR untuk Deteksi *Real-time*

Hasil eksperimen menunjukkan bahwa DETR yang dioptimasi melalui pengurangan jumlah *transformer layer* dan *object queries* dapat diterapkan secara efektif pada tugas deteksi alfabet Bahasa Isyarat Indonesia (BISINDO) berbasis citra. Dengan konfigurasi arsitektur yang lebih ringan, sistem mampu mencapai kecepatan inferensi *real-time* tanpa memerlukan varian DETR khusus seperti RT-DETR.

Meskipun performa deteksi statis pada data uji mengalami penurunan akibat penerapan augmentasi data *on-the-fly*, sistem secara keseluruhan menunjukkan peningkatan stabilitas dan keandalan pada skenario *real-time*. Hal ini menegaskan bahwa evaluasi sistem deteksi tidak dapat bergantung sepenuhnya pada metrik akurasi statis, tetapi juga perlu mempertimbangkan performa pada kondisi penggunaan langsung.

Keterbatasan penelitian ini terletak pada penggunaan dataset dengan latar belakang yang relatif terkontrol dan jumlah subjek yang terbatas. Meskipun demikian, hasil uji *real-time* pada subjek dan kondisi pencahayaan yang berbeda menunjukkan potensi pengembangan lebih lanjut untuk aplikasi yang lebih luas.

4. Kesimpulan

Penelitian ini mengkaji penerapan DETR yang dioptimasi dan dikombinasikan dengan strategi augmentasi data *on-the-fly* untuk deteksi alfabet Bahasa Isyarat Indonesia (BISINDO) berbasis citra. Hasil eksperimen menunjukkan bahwa optimasi arsitektur DETR melalui pengurangan jumlah *transformer layer* dan *object queries* mampu meningkatkan efisiensi inferensi secara signifikan, sehingga model dapat mencapai performa *real-time* hingga 58 FPS pada GPU dan 8 FPS pada CPU. Selain itu, pada evaluasi statis berbasis data uji, model tanpa augmentasi mencapai nilai AP50 sebesar 0,987, sedangkan model dengan augmentasi data *on-the-fly* memperoleh nilai AP50 sebesar 0,848.

Meskipun penerapan augmentasi data *on-the-fly* menghasilkan penurunan nilai AP50 pada evaluasi statis, model menunjukkan peningkatan stabilitas dan konsistensi deteksi pada pengujian *real-time*. Model dengan augmentasi mampu menghasilkan deteksi yang lebih andal pada input video *frame-per-frame*, termasuk pada subjek yang berbeda dan variasi kondisi pencahayaan. Temuan ini menunjukkan bahwa augmentasi data *on-the-fly* berperan penting dalam meningkatkan kemampuan generalisasi model pada kondisi penggunaan nyata, terutama pada dataset dengan jumlah terbatas dan karakteristik visual yang homogen.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa kombinasi optimasi arsitektur DETR dan augmentasi data *on-the-fly* merupakan pendekatan

yang efektif untuk pengembangan sistem deteksi alfabet BISINDO yang layak diterapkan pada skenario *real-time*. Penelitian ini menegaskan bahwa evaluasi sistem deteksi tidak dapat hanya bergantung pada metrik akurasi statis seperti AP50, tetapi juga perlu mempertimbangkan performa dan stabilitas sistem pada kondisi penggunaan langsung.

Sebagai arah penelitian selanjutnya, pengembangan dapat difokuskan pada perluasan dataset dengan melibatkan lebih banyak subjek, latar belakang, dan kondisi pencahayaan yang lebih beragam untuk meningkatkan generalisasi model. Selain itu, pengembangan menuju deteksi gestur dinamis berbasis video serta integrasi dengan sistem pengenalan bahasa isyarat tingkat kata atau kalimat menjadi peluang riset lanjutan yang potensial untuk meningkatkan kegunaan sistem dalam aplikasi nyata.

Daftar Pustaka:

- Agata, A. W., Saputra, W. S. J., & Putra, C. A. (2024). Pengenalan Bahasa Isyarat Indonesia (Bisindo) Menggunakan Algoritma Scale Invariant Feature Transform (Sift) Dan Convolutional Neural Network (Cnn). *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(1), 1054–1061. <https://doi.org/10.36040/jati.v8i1.8917>
- Agustin, R. R., Maulana, H., & Mandyartha, E. P. (2023). Detection of Actions Bisindo (Indonesian Sign Language) Into Text-To-Speech Using Long Short-Term Memory With Mediapipe Holistics. *Jurnal Teknik Informatika (Jutif)*, 5(4), 1051–1061. <https://doi.org/10.52436/1.jutif.2024.5.4.1492>
- Al-Fahrezi, M. A. (2025). Pengaruh Augmentasi Data Terhadap Akurasi Pelatihan Model CNN untuk Klasifikasi Jenis Ikan. *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, 6(2), 177–185. <https://doi.org/10.62527/jitsi.6.2.471>
- Arisandi, L., & Satya, B. (2022). Sistem Klarifikasi Bahasa Isyarat Indonesia (Bisindo) Dengan Menggunakan Algoritma Convolutional Neural Network. *Jurnal Sistem Cerdas*, 5(3), 135–146. <https://doi.org/10.37396/jsc.v5i3.262>
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. In *European Conference on Computer Vision (ECCV): 12346 LNCS* (pp. 213–229). https://doi.org/10.1007/978-3-030-58452-8_13
- Fernanda, B. A., & Bastian, A. (2025). ANALISIS KOMPARATIF YOLOV8: AUGMENTASI VS. DATA SINTETIS UNTUK DETEKSI RITEL TERBATAS. *SEMINAR TEKNOLOGI MAJALENGKA (STIMA)*, 9(1), 395–402. <https://prosiding.unma.ac.id/index.php/stima/article/view/1362>
- Fitriani, L., Sanusi, A., Rismala, R., & Tresnawati, D. (2025). Transformer-Based Detection Model for Number Recognition on Electric kWh Meters. *JUITA: Jurnal Informatika*, 13(2), 135–143. <https://doi.org/10.30595/juita.v13i2.26161>
- Khotijah, S., Juliana, J., & Driyani, D. (2023). Perancangan Media Pembelajaran Interaktif Bahasa Isyarat Bisindo Untuk Penyandang Disabilitas Tuna Rungu Berbasis Android. *Jurnal Ilmiah Multidisiplin*, 2(1), 142–149. <https://doi.org/10.59000/jim.v2i1.101>
- Maghfiroh, N. (2022). Bahasa Indonesia sebagai Alat Komunikasi Masyarakat dalam Kehidupan Sehari-hari. *Komunikologi: Jurnal Ilmiah Ilmu Komunikasi*, 19(02), 102–107. <https://doi.org/10.47007/jkomu.v19i02>
- Ricky Putra Sardika, & Widhiarso, W. (2025). Klasifikasi Otomatis Tingkat Kerusakan Retak Bangunan pada Citra Digital Menggunakan MobileNetV2 dan Augmentasi Data. *Arcitech: Journal of Computer Science and Artificial Intelligence*, 5(1), 108–124. <https://doi.org/10.29240/arcitech.v5i1.13938>
- Sari, I., Fivrenodi, Altiarika, E., & Sarwindah. (2023). Sistem Pengembangan Bahasa Isyarat Untuk Berkomunikasi dengan Penyandang Disabilitas (Tunarungu). *Journal of Information Technology and Society*, 1(1), 20–25. <https://doi.org/10.35438/jits.v1i1.21>
- Subkhi, M. B., Trinurais, M. Y., Wibowo, R. K. A., & Prakosa, B. R. (2024). Deteksi Bahasa Isyarat Berdasarkan SIBI (Sistem Bahasa Isyarat) menggunakan Transfer Learning. *Seminar Nasional Teknologi & Sains*, 3(1), 361–369. <https://doi.org/10.29407/stains.v3i1.4347>
- Suciati, A., Sari, D. K., Yunus, A. P., & Amaliah, N. R. (2025). Detection of Vulgarity in Anime Character: Implementation of Detection Transformer. *JURNAL TEKNIK INFORMATIKA*, 18(1), 157–165. <https://doi.org/10.15408/jti.v18i1.46064>
- Trisianto, D., & Limantara, M. A. (2024). Sistem Pembelajaran Isyarat Bahasa Indonesia (Sibi) Menggunakan Metode Convolutional Neural Network (Cnn). *Jurnal Sistem Cerdas Dan Rekayasa (JSCR)*, 6(2), 2656–7504. <https://doi.org/10.61293/jscr.v6i2.735>
- Ubaidillah, R. F., Sulistiyo, M. D., Kosala, G., Rachmawati, E., & Haryadi, D. (2025). Advancing Vehicle Logo Detection with DETR to Handle Small Logos and Low-Quality Images. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 9(4), 796–804. <https://doi.org/10.29207/resti.v9i4.6236>
- Ulfah, S. M. (2023). Penerapan Bahasa Isyarat dalam Pembelajaran bagi Anak Berkebutuhan Khusus Tuna Rungu. *Journal of Disability Studies and Research*, 2(1), 29–42. <https://doi.org/10.30631/jdsr.v2i1.1764>
- Yisti Vita Via, Wahyu S. J. Saputra, Mohammad Idham Fachrurrozi, Eva Yulia Puspaningrum, Fetty Tri Anggraeny, & Salamun Rohman

Nudin. (2023). Object Localization and Detecting Alphabet in Sign Language BISINDO Using Convolution Neural Network. *Technium: Romanian Journal of Applied Sciences and Technology*, 16(1), 143–149. <https://doi.org/10.47577/technium.v16i.9973>

Yu, L., Tang, L., & Mu, L. (2025). A Review of DEtection TRansformer: From Basic Architecture to Advanced Developments and Visual Perception Applications. *Sensors*, 25(13), 3952. <https://doi.org/10.3390/s25133952>

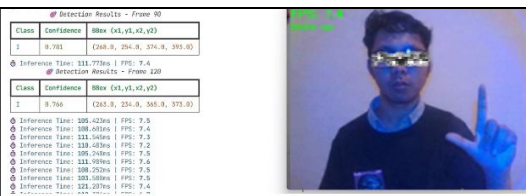
Lampiran A. Contoh Hasil Deteksi Real-Time pada Hasil Model Skenario 1



Gambar 13. Gagal mendeteksi alfabet B



Gambar 14. Gagal mendeteksi alfabet H



Gambar 15. Gagal mendeteksi alfabet L pada kondisi pencahayaan berbeda



Gambar 16. Gagal mendeteksi alfabet S pada kondisi pencahayaan berbeda



Gambar 17. Berhasil mendeteksi alfabet A



Gambar 18. Berhasil mendeteksi alfabet G

Lampiran B. Contoh Hasil Deteksi Real-Time pada Hasil Model Skenario 2



Gambar 19. Berhasil mendeteksi alfabet J



Gambar 20. Berhasil mendeteksi alfabet J pada subjek berbeda



Gambar 21. Berhasil mendeteksi alfabet K pada subjek berbeda



Gambar 22. Berhasil mendeteksi alfabet K pada kondisi pencahayaan berbeda



Gambar 23. Berhasil mendeteksi alfabet M pada kondisi pencahayaan berbeda