

Analisis Ketahanan Model ResNet-50 pada Klasifikasi Bahasa Isyarat Arab terhadap Degradasi Citra Bawah Air

Muhammad Ilham*, Aris Rakhmadi

Fakultas Komunikasi dan Informatika, Program Studi Teknik Informatika, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia

Email: ¹*its.haemml@gmail.com, ²aris.rakhmadi@ums.ac.id

Email Penulis Korespondensi: its.haemml@gmail.com

Submitted: 04/03/2026; Accepted: 19/03/2026; Published: 19/03/2026

Abstrak—Pengenalan bahasa isyarat otomatis menggunakan *deep learning*, khususnya *Convolutional Neural Networks* (CNN), telah menunjukkan potensi besar. Arsitektur ResNet, melalui *transfer learning*, sering dilaporkan mencapai akurasi tinggi untuk klasifikasi Bahasa Isyarat Alfabet Arab dalam kondisi ideal. Namun, ketahanan (*robustness*) model terhadap distorsi visual di dunia nyata masih menjadi tantangan signifikan yang kurang dieksplorasi. Penelitian ini bertujuan mengembangkan model klasifikasi berbasis ResNet-50 sekaligus menganalisis ketahanannya secara komprehensif. Kontribusi utama dari penelitian ini adalah memetakan sejauh mana batas toleransi dan tingkat penurunan performa arsitektur ResNet saat menghadapi degradasi citra. Evaluasi dilakukan pada data uji ideal dan data yang dimodifikasi untuk mensimulasikan efek visual bawah air. Simulasi bawah air ini dipilih sebagai skenario uji stres ekstrem karena secara teknis mampu merepresentasikan akumulasi gangguan optik dunia nyata secara bersamaan, seperti penurunan kontras, kekeruhan, dan pembiasan cahaya. Hasil evaluasi kuantitatif menunjukkan bahwa model berkinerja sangat baik dengan akurasi mencapai 96,95% pada kondisi ideal. Namun, paparan distorsi bawah air mengakibatkan penurunan akurasi sebesar 4,24% menjadi 92,71%. Meskipun terjadi reduksi performa yang nyata, model tetap mampu mempertahankan *F1-Score* di angka 92,79%. Temuan ini memberikan bukti empiris mengenai batas kemampuan arsitektur ResNet saat menghadapi degradasi visual, sekaligus menegaskan pentingnya pengujian *robustness* sebelum model diterapkan secara andal di lingkungan *non-ideal* yang penuh ketidakpastian visual.

Kata Kunci: Pengenalan Bahasa Isyarat; Convolutional Neural Network (CNN); ResNet; Transfer Learning; Ketahanan Model; Distorsi Visual

Abstract—Automatic sign language recognition using deep learning, particularly Convolutional Neural Networks (CNNs), has shown significant potential. The ResNet architecture, through transfer learning, is frequently reported to achieve high accuracy for Arabic Sign Language Alphabet classification under ideal conditions. However, the robustness of these models against real-world visual distortions remains a significant, yet under-explored challenge. This research aims to develop a ResNet-50-based classification model while comprehensively analyzing its robustness. The primary contribution of this research is mapping the tolerance limits and the extent of performance degradation of the ResNet architecture when facing image degradation. Evaluation was conducted on both ideal test data and test data digitally modified to simulate underwater visual effects. This underwater simulation was selected as an extreme stress test scenario because it technically represents an accumulation of simultaneous real-world optical distortions, such as contrast reduction, turbidity (haziness), and light refraction. Quantitative evaluation results show that the model performs excellently with an accuracy of 96.95% under ideal conditions. However, exposure to underwater distortion resulted in an accuracy drop of 4.24%, reducing it to 92.71%. Despite this noticeable performance reduction, the model maintained an F1-Score of 92.79%. These findings provide empirical evidence regarding the capability limits of the ResNet architecture when facing visual degradation, while also emphasizing the importance of robustness testing before deep learning models can be reliably deployed in non-ideal environments full of visual uncertainties.

Keywords: Sign Language Recognition; Convolutional Neural Network (CNN); ResNet; Transfer Learning; Model Robustness; Visual Distortion

1. PENDAHULUAN

Komunikasi visual melalui bahasa isyarat merupakan jembatan fundamental bagi jutaan individu penyandang tunarungu dan tunawicara untuk berinteraksi dengan dunia sosial di sekitar mereka. Namun, kesenjangan komunikasi sering terjadi karena sebagian besar masyarakat umum tidak memahami bahasa ini, sehingga menuntut kehadiran teknologi pengenalan otomatis sebagai instrumen penerjemah. Bahasa isyarat sendiri adalah bentuk komunikasi visual yang kompleks, di mana proses penyampaiannya sangat mengandalkan koordinasi gerak tubuh, terutama detail pada isyarat tangan dan variasi ekspresi wajah [1]. Oleh karena itu, seiring dengan pesatnya kemajuan teknologi informasi, pengembangan model klasifikasi otomatis berbasis *Deep Learning* hadir sebagai solusi potensial untuk menerjemahkan isyarat ini secara *real-time*, yang diharapkan dapat meningkatkan inklusivitas baik dalam sektor pendidikan maupun layanan publik.

Penelitian ini secara khusus berfokus pada Bahasa Isyarat Alfabet Arab (Hijaiyah) yang memiliki kompleksitas tersendiri dibandingkan alfabet Latin. Representasi huruf Arab dalam bahasa isyarat meliputi bentuk jari tangan yang unik dan presisi untuk membedakan huruf 'alif' sampai 'ya', serta ditambah dengan beberapa isyarat lain yang penting dan memiliki makna tersendiri, contohnya karakter khusus seperti 'alif lam' dan 'ta marbutah' [2]. Untuk menerjemahkan kompleksitas visual ini ke dalam sistem komputer, pemodelan menggunakan arsitektur *Deep Learning*, khususnya *Convolutional Neural Networks* (CNN), telah menjadi standar utama. Tinjauan literatur sistematis dalam satu dekade terakhir menunjukkan adanya pergeseran tren yang kuat dari metode *machine learning* konvensional menuju *Deep Learning* dalam pengenalan bahasa isyarat Arab [3]. Model berbasis CNN, termasuk

pemanfaatan arsitektur seperti ResNet, terbukti mendominasi penelitian terkini karena kemampuan superiornya dalam mengekstraksi fitur spasial yang kompleks dari data visual secara otomatis [3]. Implementasi model-model tingkat lanjut inilah yang membuka jalan bagi pengembangan sistem penerjemah isyarat Arab yang berakurasi tinggi.

Kompleksitas pengenalan pola ini menuntut pemilihan arsitektur model yang tepat dan krusial. Penelitian ini memilih menggunakan ResNet (*Residual Network*) sebagai arsitektur utama. Sebagai salah satu evolusi dari *Convolutional Neural Networks* (CNN), ResNet didesain untuk mengatasi masalah degradasi pada jaringan yang sangat dalam [4]. Kuncinya terletak pada penerapan *residual learning*, di mana koneksi pintas (*shortcut connection*) ditambahkan untuk melompati beberapa lapisan dalam jaringan, memungkinkan aliran informasi yang lebih efisien selama proses pelatihan [5]. Lebih spesifik lagi, penggunaan model *pre-trained* (pralatih) ResNet-50 dinilai sangat strategis karena arsitektur ini telah terbukti menawarkan tingkat akurasi yang lebih unggul dibandingkan dengan teknik pembelajaran mesin lainnya dalam berbagai tugas klasifikasi citra [6].

Meskipun teknologi ini menjanjikan, tantangan besar yang sering terabaikan dalam implementasi model di dunia nyata adalah masalah ketahanan atau *robustness*. Banyak penelitian sebelumnya berhasil membangun model dengan akurasi tinggi, namun model tersebut sering kali dilatih dan diuji pada data citra yang ideal dan bersih. Akibatnya, sistem mengalami penurunan kinerja drastis ketika dihadapkan pada kondisi visual lingkungan nyata yang tidak terduga. Hal ini sejalan dengan temuan tinjauan sistematis oleh Alayed (2024) yang melaporkan bahwa meskipun banyak studi *Deep Learning*, khususnya ResNet, mencapai akurasi sangat tinggi hingga 100% untuk pengenalan alfabet Bahasa Isyarat Arab, penelitian-penelitian tersebut memiliki kelemahan fundamental [3]. Dataset yang digunakan disimpulkan kurang memiliki keragaman visual, baik dari segi latar belakang, pencahayaan, maupun sudut kamera, sehingga membatasi kemampuan model untuk tetap tangguh (*robust*) dalam aplikasi dunia nyata [7].

Masalah kerentanan ini diperkuat oleh literatur teknis lainnya. Wang dkk. (2021) menyatakan bahwa akurasi *Deep Convolutional Neural Networks* (DCNN) generik dapat sangat dipengaruhi oleh distorsi citra pada data masukan; penambahan sedikit distorsi saja pada set pengujian biasanya menghasilkan penurunan akurasi klasifikasi jaringan yang signifikan [8]. Artinya, model yang tampak pintar di laboratorium bisa gagal total saat diterapkan di lapangan. Faktanya, penelitian oleh Putra dan Wardhana (2015) menunjukkan bahwa sebagian besar penelitian yang ada memang masih terbatas pada penggunaan ukuran dataset yang kecil dan belum secara komprehensif menguji ketahanan model terhadap variasi latar belakang, pencahayaan, maupun orientasi tubuh pengguna [9].

Merespon celah penelitian yang ada, muncul urgensi untuk melakukan *Gap Analysis* melalui pengujian stres pada model. Salah satu cara paling efektif untuk menguji batas kemampuan model adalah dengan memberikan gangguan visual yang ekstrem [10]. Penelitian ini menggunakan simulasi efek visual bawah air sebagai studi kasus untuk distorsi ekstrem tersebut. Lingkungan bawah air memiliki karakteristik optik yang unik dan menantang. Akibat efek penyerapan yang mengurangi energi cahaya dan hamburan yang mengubah arah rambatnya, cahaya di bawah air mengalami redaman signifikan yang mengakibatkan visibilitas gambar menjadi buruk, ditandai dengan penampilan berkabut, degradasi kontras, dan objek jauh yang tampak samar [11]. Kondisi degradasi visual ini dipilih karena secara teknis mampu merepresentasikan akumulasi gangguan visual dunia nyata secara bersamaan. Sebagai contoh, efek hamburan dan berkabut mensimulasikan kualitas tangkapan kamera *webcam* beresolusi rendah atau lensa yang kotor, sementara degradasi kontras secara ketat mewakili kondisi pencahayaan dunia nyata yang ekstrem, seperti efek *backlight* atau ruangan minim cahaya (*low-light*). Oleh karena itu, jika model mampu mempertahankan kinerjanya dalam kondisi terdegradasi ini, maka model tersebut terbukti memiliki ketahanan (*robustness*) yang kuat terhadap berbagai gangguan visual saat diimplementasikan di lapangan.

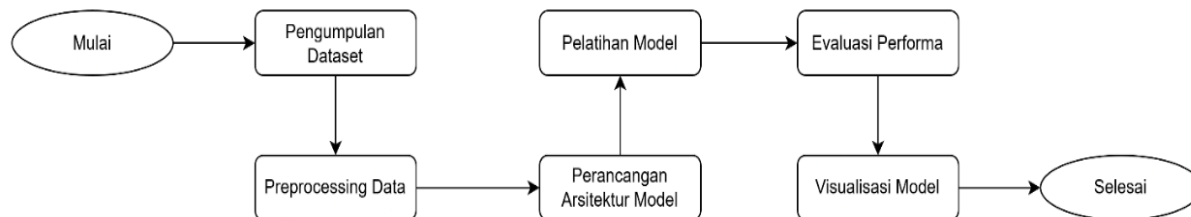
Berdasarkan uraian permasalahan dan analisis kesenjangan penelitian (*research gap*) yang telah dipaparkan sebelumnya, penelitian ini bertujuan untuk mengembangkan sebuah model klasifikasi *Deep Learning* yang presisi untuk pengenalan alfabet bahasa isyarat Arab. Pengembangan sistem tersebut menggunakan arsitektur ResNet-50 berbasis *transfer learning*, yang disertai dengan evaluasi mendalam terhadap aspek ketahanannya (*robustness*). Skenario evaluasi ini mencakup komparasi metrik kinerja antara *dataset* orisinal yang ideal dan *dataset* yang telah terdistorsi secara visual melalui simulasi bawah air. Pengujian ini dirancang secara spesifik guna memetakan batasan kemampuan (*boundary limits*) arsitektur ResNet-50 saat menghadapi degradasi kualitas citra yang ekstrem. Luaran dari penelitian ini diharapkan dapat memberikan kontribusi signifikan, baik secara teoretis maupun praktis, guna memvalidasi keandalan operasional sistem cerdas. Validasi tersebut menjadi fondasi awal yang krusial sebelum model diimplementasikan secara luas pada lingkungan *non-ideal* di dunia nyata yang memiliki tingkat ketidakpastian visual tinggi.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Pendekatan eksperimental kuantitatif diterapkan dalam penelitian ini sebagai kerangka utama untuk mengklasifikasikan citra bahasa isyarat alfabet Arab menggunakan arsitektur *Convolutional Neural Network* (CNN). Secara spesifik, pemodelan ini dibangun berbasis ResNet-50. Sebagai sebuah alur penyelesaian masalah, rangkaian tahapan operasional dirancang secara sistematis dan berurutan. Tahapan tersebut diawali dari proses pengumpulan *dataset*, prapemrosesan data, perancangan arsitektur model, pelaksanaan proses pelatihan, skenario evaluasi kinerja,

hingga tahap akhir berupa visualisasi model. Alur kerja dari keseluruhan tahapan metodologi tersebut diilustrasikan secara runut pada Gambar 1 berikut.



Gambar 1. Diagram alir tahapan metodologi penelitian, yang meliputi pengumpulan data, perancangan dan pelatihan model CNN, evaluasi performa hingga visualisasi model.

2.2 Pengumpulan Dataset

Penelitian ini menggunakan dataset publik dari Kaggle (Albarham, 2023) yang berisi 7.855 citra alfabet isyarat Arab dengan berbagai variasi visual. Struktur direktorinya telah kompatibel dengan format ImageFolder PyTorch, sehingga proses pelabelan dapat dilakukan secara otomatis mengikuti nama folder kelas. Distribusi data selengkapnya dirincikan pada Tabel 1.

Tabel 1. Rincian distribusi data citra pada setiap kelas alfabet bahasa isyarat Arab

No	Class	Total data	No	Class	Total data
	<i>Ain</i>	244		<i>Noon</i>	237
	<i>Al</i>	276		<i>Qaf</i>	219
	<i>Alef</i>	287		<i>Reh</i>	227
	<i>Beh</i>	307		<i>Sad</i>	270
	<i>Dad</i>	266		<i>Seen</i>	266
	<i>Dal</i>	235		<i>Sheen</i>	278
	<i>Feh</i>	255		<i>Tah</i>	226
	<i>Ghain</i>	230		<i>Teh</i>	311
	<i>Hah</i>	246		<i>Teh_Marbuta</i>	257
	<i>Heh</i>	253		<i>Thal</i>	202
	<i>Jeem</i>	210		<i>Theh</i>	305
	<i>Kaf</i>	264		<i>Waw</i>	249
	<i>Khah</i>	250		<i>Yeh</i>	271
	<i>Laa</i>	268		<i>Zah</i>	232
	<i>Lam</i>	260		<i>Zain</i>	201
	<i>Meem</i>	253			

2.3 Preprocessing Data

Preprocessing adalah proses yang dilakukan untuk mentransformasi data sebelum data tersebut digunakan sebagai informasi pembelajaran model [12] Tujuan dari proses ini adalah untuk memastikan kualitas dan konsistensi data sehingga pelatihan model dapat berjalan secara optimal [13]. tahapan preprocessing data disini meliputi Pembagian Dataset, Transformasi Data, Dan Augmentasi data.

a. Pembagian dataset

Dataset dibagi menjadi tiga bagian: 70% data *training*, 15% data *validation* untuk pemantauan dan pencegahan *overfitting*, serta 15% data *test* untuk evaluasi final.

b. Transformasi data

Proses standarisasi gambar menggunakan `torchvision.transform`. dimana mencakup mengubah ukuran gambar menjadi input standar ResNet yaitu (224x224 pixel), mengubah gambar dari format PIL Image menjadi format Tensor PyTorch, melakukan standarisasi pada gambar yang sudah diubah menjadi Tensor menggunakan nilai mean dan std untuk setiap channel (saluran warna) dari gambar dan menerapkan rumus matematika berikut

$$v_{norm} = \frac{v - \mu}{\sigma} \quad (1)$$

c. Augmentasi data

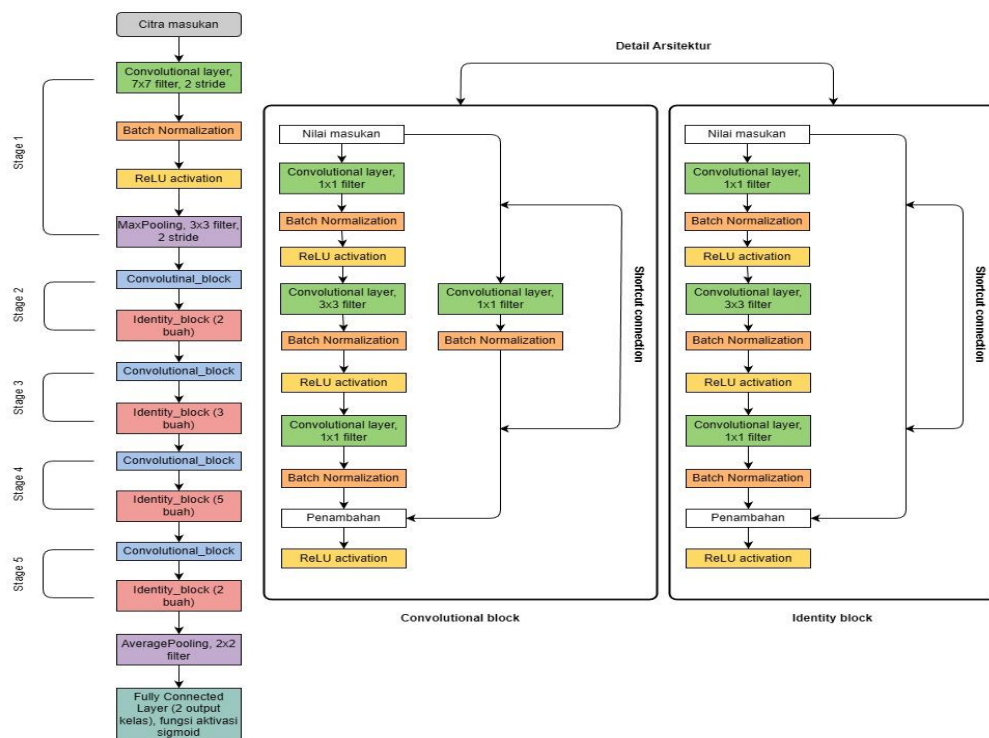
Proses augmentasi data dilakukan dengan menerapkan (rotasi acak $\pm 15^\circ$ dan flip horizontal 50%) untuk menambah variasi dan membuat model lebih tangguh, sehingga tetap mengenali isyarat huruf yang miring atau terbalik.

2.4 Perancangan Arsitektur Model

Model klasifikasi bahasa isyarat alfabet arab dibangun menggunakan arsitektur *deep learning* yaitu Convolutional Neural Network (CNN). Secara spesifik arsitektur yang dipilih adalah ResNet-50. Alasan pemilihan arsitektur ResNet

adalah karena kinerjanya yang telah terbukti sangat baik dalam berbagai kompetisi klasifikasi citra. Keunggulan utamanya terletak pada penggunaan *residual connections* (koneksi residual), yang memungkinkannya untuk melatih jaringan yang sangat dalam secara efektif tanpa mengalami masalah *vanishing gradient* (melemahnya gradien). Berdasarkan pernyataan oleh Banu dan Hanggoro (2025) arsitektur ResNet dan VGG terbukti berkinerja sangat baik untuk klasifikasi gambar kompleks, khususnya jika didukung sumber daya komputasi besar (server/cloud) untuk akurasi tinggi [14].

Studi pustaka yang dilakukan oleh Nashrullah dkk. (2020) memaparkan bahwa ResNet-50 merupakan arsitektur CNN yang memperkenalkan *shortcut connections*. Arsitektur ResNet-50 dirancang untuk mengatasi *vanishing gradient problem*, Solusi ini memungkinkan pelatihan jaringan (network) yang sangat dalam secara efektif, yang biasanya gagal atau mengalami penurunan akurasi jika hanya menambah lapisan konvensional [15]. *Shortcut connections* pada ResNet diperkenalkan sebagai solusi untuk masalah fundamental ini dimana dapat di gambarkan pada Gambar 2.



Gambar 2. Diagram blok arsitektur ResNet-50. Sumber: [15]

Arsitektur ResNet-50, seperti yang diilustrasikan pada Gambar 2, memperkenalkan *shortcut connections* untuk mengatasi masalah degradasi fitur. Konsep ini menambahkan input dari layer sebelumnya ke output-nya, memastikan fitur-fitur penting tidak hilang seiring bertambahnya kedalaman jaringan. Terdiri dari 5 stage konvolusi, diikuti oleh average pooling, dan diakhiri fully connected layer sebagai prediktor.

Penelitian ini tidak melatih model dari nol (from scratch), melainkan menerapkan metode transfer learning melalui teknik *fine-tuning*. Pendekatan ini dipilih untuk memanfaatkan pengetahuan yang telah dimiliki oleh model yang sudah dilatih pada dataset berskala besar, sehingga tidak perlu melatih model dari awal [16]. Proses fine-tuning ini terdiri dari dua langkah utama

a. Memuat bobot *pre-trained*

Langkah pertama adalah memuat arsitektur ResNet-50 yang telah dilatih sebelumnya (pretrained=true) pada dataset ImageNet. Dataset ini berisi lebih dari satu juta gambar dari 1000 kelas yang berbeda (seperti hewan, kendaraan, dan objek sehari-hari). Dengan memuat bobot ini, model telah memiliki kemampuan dasar untuk mengenali berbagai fitur visual, seperti tepi, tekstur, dan bentuk.

b. Modifikasi Lapisan Klasifikasi Akhir

Karena model asli dilatih untuk 1000 kelas ImageNet, lapisan klasifikasi akhirnya (dikenal sebagai fully connected layer atau model.fc) harus dimodifikasi. Lapisan ini diganti dengan lapisan nn.Linear baru yang memiliki jumlah output yang sesuai dengan jumlah kelas dalam penelitian ini, yaitu 31 kelas Bahasa Isyarat Alfabet Arab. Dengan demikian, model "diadaptasi" dari tugas klasifikasi umum menjadi tugas klasifikasi spesifik bahasa isyarat.

2.5 Pelatihan Model

Tahapan pelatihan model akan di jalankan pada lingkungan komputasi yang spesifik dan diatur oleh serangkaian *hyperparameter* untuk mengendalikan jalannya pembelajaran

a. Lingkungan Pelatihan

Proses pelatihan dan evaluasi model dilakukan menggunakan perangkat keras laptop yang dilengkapi dengan *Graphics Processing Unit* (GPU) NVIDIA GeForce GTX 1650 Ti, dengan seluruh alur eksperimen dijalankan dalam lingkungan Jupyter Notebook. Penggunaan GPU esensial untuk mengakselerasi komputasi yang intensif dalam pelatihan deep learning. Sejalan dengan praktik umum yang memanfaatkan akselerasi perangkat keras seperti penggunaan komputasi awan Google ber-GPU yang dilaporkan dapat mempersingkat waktu pelatihan seperti yang dilaporkan oleh Imaduddin dkk. (2022) dan Rakhmadi dkk. (2025) [17], [18]. Seluruh eksperimen diimplementasikan menggunakan bahasa pemrograman Python dengan *library* utama PyTorch, yang merupakan sebuah *framework open-source*. *framework* ini populer untuk *deep learning* karena menawarkan fleksibilitas yang tinggi dan mudah digunakan, khususnya saat merancang dan mengevaluasi model pembelajaran mendalam [19].

b. Konfigurasi Hyperparameter

Pengoptimalkan proses pelatihan diperlukan dalam pembelajaran model, Sehingga beberapa *hyperparameter* kunci akan ditetapkan seperti pada tabel 2 berikut:

Tabel 2. Konfigurasi *hyperparameter* untuk pelatihan model

Hyperparameter	Nilai/Algoritma
Loss Function	CrossEntropyLoss
Optimizer	Adam
Learning Rate	0.001
Epoch	30
Batch Size	32

loss function yang digunakan adalah *CrossEntropyLoss*. Fungsi ini dipilih karena merupakan standar industri untuk masalah klasifikasi multi-kelas (ketika terdapat lebih dari dua kelas target). *CrossEntropyLoss* secara efektif mengukur seberapa jauh perbedaan antara hasil prediksi probabilitas model dengan label yang sebenarnya.

Pembaruan bobot model dilakukan menggunakan algoritma optimasi Adam. Adam, yang merupakan ekstensi dari Stochastic Gradient Descent (SGD), digunakan karena kemampuannya yang adaptif dalam menyesuaikan learning rate untuk setiap parameter model, sehingga seringkali mencapai konvergensi lebih cepat dan lebih efisien [20]. *Learning Rate* awal ditetapkan sebesar 0.001, yang menentukan seberapa besar langkah yang diambil oleh optimizer untuk memperbarui bobot model pada setiap iterasi.

Model dilatih selama 30 Epoch. Satu epoch menandakan satu siklus penuh di mana model telah melihat dan belajar dari keseluruhan data latih sebanyak satu kali. Terakhir, *Batch Size* diatur ke 32. Ini berarti data latih tidak diproses sekaligus, melainkan dibagi menjadi kelompok-kelompok kecil (*batch*) yang terdiri dari 32 gambar. Model memperbarui bobotnya setelah memproses setiap *batch*, di mana penggunaan *batch size* ini membantu dalam mengelola penggunaan memori GPU dan menstabilkan proses pembelajaran.

2.6 Skenario Evaluasi Performa

Tahap ke-5 dari penelitian ini adalah evaluasi kinerja model secara kuantitatif. Tahap ini dirancang untuk mengukur dua aspek: akurasi baseline model pada data ideal dan ketahanan (*robustness*) model terhadap distorsi visual.

a. Pengukuran performa model, digunakan beberapa metrik standar dalam *machine learning* yang disediakan oleh *library* Scikit-learn [21]. Metrik-metrik tersebut adalah:

Accuracy: Metrik utama yang mengukur persentase total prediksi yang benar dari keseluruhan data uji.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Precision: Mengukur probabilitas ketepatan model saat memprediksi kelas positif

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

Recall: mengukur proporsi data aktual dari kelas positif yang berhasil diidentifikasi secara tepat oleh model.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

F1-Score: merupakan nilai rata-rata harmonik (harmonic mean) antara presisi dan recall.

$$\text{F1 - Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

Confusion matrix: matriks tabulasi silang yang membandingkan hasil prediksi model dengan label sebenarnya.

b. Skenario Pengujian Model

1. Pengujian Skenario 1: Kinerja Kondisi Ideal Pengujian pertama dilakukan pada seluruh data uji (15%) dalam kondisi aslinya (bersih). Tujuan dari skenario ini adalah untuk mendapatkan *baseline* kinerja model. Hasil dari pengujian ini menunjukkan seberapa baik model dapat menggeneralisasi pengetahuannya pada data baru yang memiliki karakteristik serupa dengan data latih (kondisi ideal).
2. Pengujian Skenario 2: Analisis Ketahanan (Distorsi Bawah Air) Pengujian kedua dirancang untuk mengukur ketahanan model. Pada skenario ini, data uji yang sama (15%) dari Skenario 1 akan diberi transformasi kustom

"Efek Underwater". Transformasi ini dibuat menggunakan *library* OpenCV untuk menambahkan filter warna biru/hijau dan *gaussian blur* yang mensimulasikan kondisi visual terdistorsi di bawah air.

2.7 Visualisasi Implementasi Model

Demonstrasi kinerja dan analisis perilaku model ResNet yang telah dilatih diwujudkan melalui sebuah prototipe aplikasi web interaktif. Aplikasi ini tidak hanya menyajikan hasil akhir (prediksi), tetapi juga memvisualisasikan proses internal model, sehingga berfungsi sebagai *proof-of-concept* dari penelitian yang dilakukan.

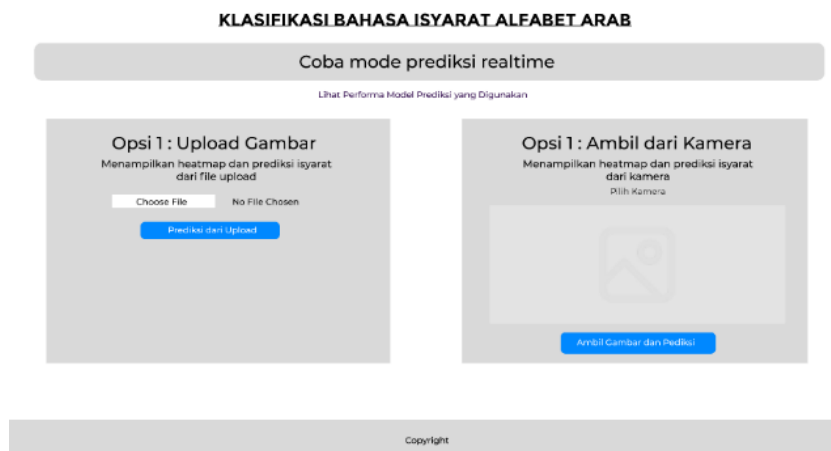
Web framework yang akan digunakan dalam membangun webnya adalah Flask, sebuah *micro-framework* Python yang simple dan flexibel [22] dan juga cocok untuk proyek kecil serta mempermudah pengembang dalam menggunakannya karena fiturnya yang sederhana [23]. Untuk menangani fungsionalitas prediksi *real-time*, aplikasi ini diintegrasikan dengan *library* Flask-SocketIO. SocketIO memungkinkan komunikasi dua arah (*bi-directional*) yang instan antara server (Python) dan klien (browser pengguna) melalui WebSocket [24], sehingga data prediksi dapat dikirim secara terus-menerus tanpa perlu *me-refresh* halaman

a. Fitur utama aplikasi

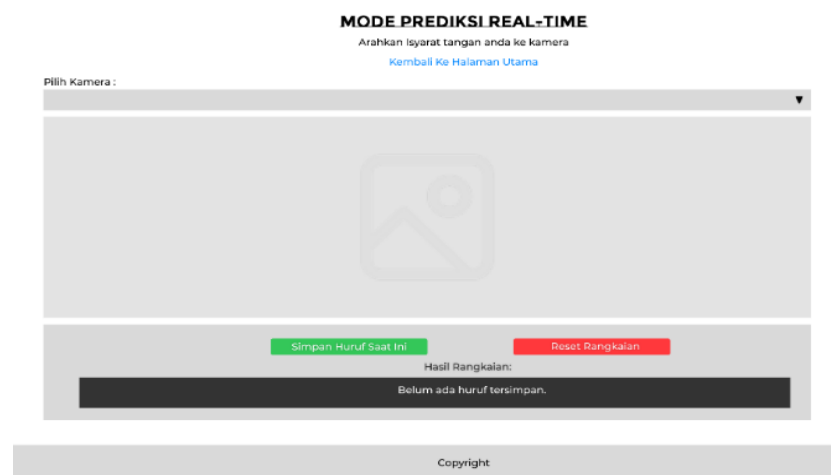
Prototipe aplikasi yang dikembangkan memiliki dua fungsionalitas utama, yaitu mode analisis *snapshot* dan mode prediksi *real-time*. Pada mode analisis *snapshot*, sistem memfasilitasi pengguna untuk memproses citra statis guna menghasilkan komparasi visual secara berdampingan antara kondisi citra ideal dan citra terdistorsi bawah air. Mode ini secara spesifik mengintegrasikan visualisasi *heatmap* Grad-CAM untuk mengevaluasi dampak degradasi citra terhadap pergeseran area fokus (atensi) model secara interpretatif. Sementara itu, mode prediksi *real-time* difungsikan untuk menganalisis aliran video (*video stream*) secara kontinu dari perangkat kamera pengguna. Keluaran dari mode ini berupa *overlay* yang memuat label kelas prediksi beserta nilai probabilitasnya (*confidence level*) secara langsung pada antarmuka video. Guna mendukung pengujian praktis, antarmuka pada mode ini juga dibekali dengan konfigurasi pemilihan sumber kamera serta fitur manipulasi ("Simpan Huruf" dan "Reset") yang memungkinkan pengguna untuk merangkai karakter hasil prediksi menjadi satu kesatuan kata.

b. Mockup aplikasi

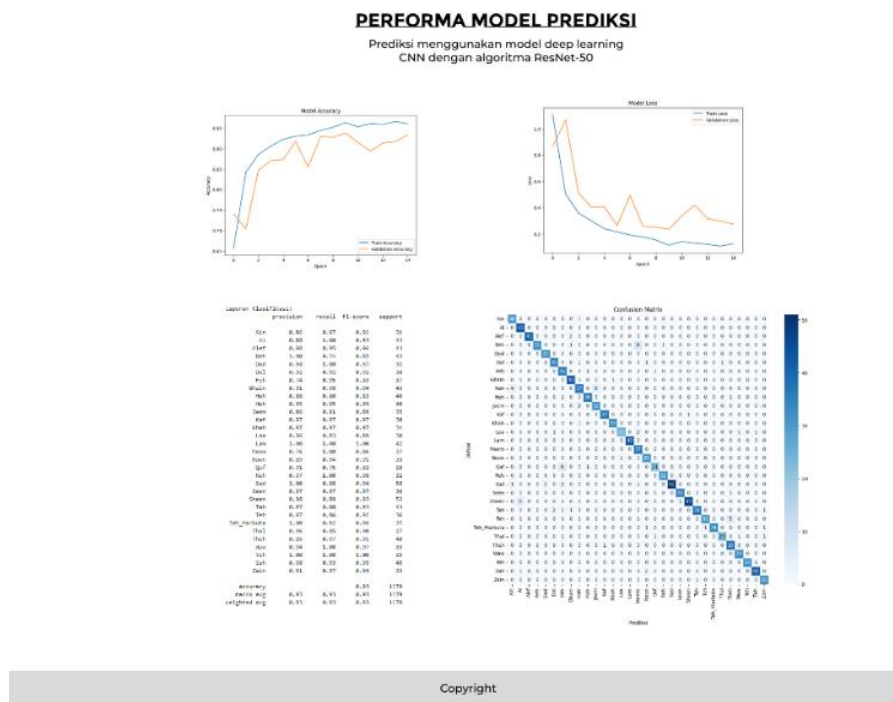
Tampilan mockup dari aplikasi berbasis web yang akan dibuat digambarkan pada gambar 3, 4 dan 5 berikut :



Gambar 3. Rancangan antarmuka utama (*dashboard*) aplikasi web interaktif pengenalan bahasa isyarat.



Gambar 4. Rancangan antarmuka mode prediksi secara *real-time* berbasis tangkapan kamera langsung.



Gambar 5. Rancangan antarmuka halaman laporan visualisasi kinerja dan metrik evaluasi model ResNet-50.

3. HASIL DAN PEMBAHASAN

3.1 Pembagian data

Penelitian ini mendistribusikan dataset ke dalam tiga subset independen : data latih (training), data validasi (validation), dan data uji (testing) dengan rasio 70:15:15. Teknik pembagian yang digunakan adalah Stratified Random Sampling untuk memastikan proporsi setiap kelas target tersebar merata dan seimbang di seluruh subset data.

Proses partisi data dilakukan secara bertahap menggunakan algoritma `train_test_split` dari pustaka Scikit-learn dengan parameter `random_state=42` untuk menjamin konsistensi hasil eksperimen (reproducibility). Rincian distribusi data dapat dilihat pada Gambar 6 berikut:

```

--- Detail Distribusi Data Per Kelas (Stratified) ---
Kelas  Latih (70%)  Validasi (15%)  Uji (15%)  Total
Ain      171             37              36         244
Al       193             41              42         276
Alef     201             43              43         287
Beh      215             46              46         307
Dad      186             40              40         266
Dal      165             35              35         235
Feh      178             38              39         255
Ghain    161             35              34         230
Hah      172             37              37         246
Heh      177             38              38         253
Jeem     147             32              31         210
Kaf      185             39              40         264
Khah     175             38              37         250
Laa      188             40              40         268
Lam      182             39              39         260
Meem     177             38              38         253
Noon     166             36              35         237
Qaf      153             33              33         219
Reh      159             34              34         227
Sad      189             40              41         270
Seen     186             40              40         266
Sheen    195             41              42         278
Tah      158             34              34         226
Teh      218             46              47         311
Teh_Marbuta 180         38              39         257
Thal     141             31              30         202
Theh     213             46              46         305
Waw      174             38              37         249
Yeh      190             40              41         271
Zah      162             35              35         232
Zain     141             30              30         201
TOTAL    5498           1178           1179       7855
    
```

Gambar 6. Rincian distribusi dataset ke dalam subset data latih (70%), validasi (15%), dan uji (15%) pada setiap kelas.

3.2 Preprocessing data

Tahap preprocessing data dilakukan menggunakan modul torchvision.transforms untuk menstandarisasi input sebelum diproses oleh model. Seluruh citra dalam dataset, baik data latih, validasi, maupun uji, mengalami penyeragaman dimensi (resizing) menjadi 224 x 224 piksel. Selanjutnya, citra dikonversi ke format Tensor dan dinormalisasi menggunakan nilai rata-rata (mean) [0.485, 0.456, 0.406] serta standar deviasi (std) [0.229, 0.224, 0.225] mengikuti standar statistik dataset ImageNet.

Perbedaan perlakuan diterapkan khusus pada subset data latih (*training set*) dengan mengimplementasikan teknik augmentasi data, yaitu rotasi acak (*random rotation*) maksimal 15° dan pembalikan horizontal secara acak (*random horizontal flip*). Penerapan *horizontal flip* dinilai sangat relevan dan aman untuk dataset bahasa isyarat abjad. Transformasi ini secara teknis melatih model agar memiliki sifat invarian terhadap tangan dominan pengguna (baik tangan kanan maupun kiri). Pada konteks gestur alfabet Arab yang bersifat statis, representasi spasial cermin (*mirrored spatial representation*) tidak merusak struktur kelas maupun mengubah makna semantik dari huruf tersebut. Secara keseluruhan, augmentasi ini bertujuan untuk memperkaya variasi data dan mencegah *overfitting*, sedangkan data validasi dan uji tidak melalui proses augmentasi untuk menjaga objektivitas evaluasi. Rincian prapemrosesan ini disajikan pada Tabel 3 berikut..

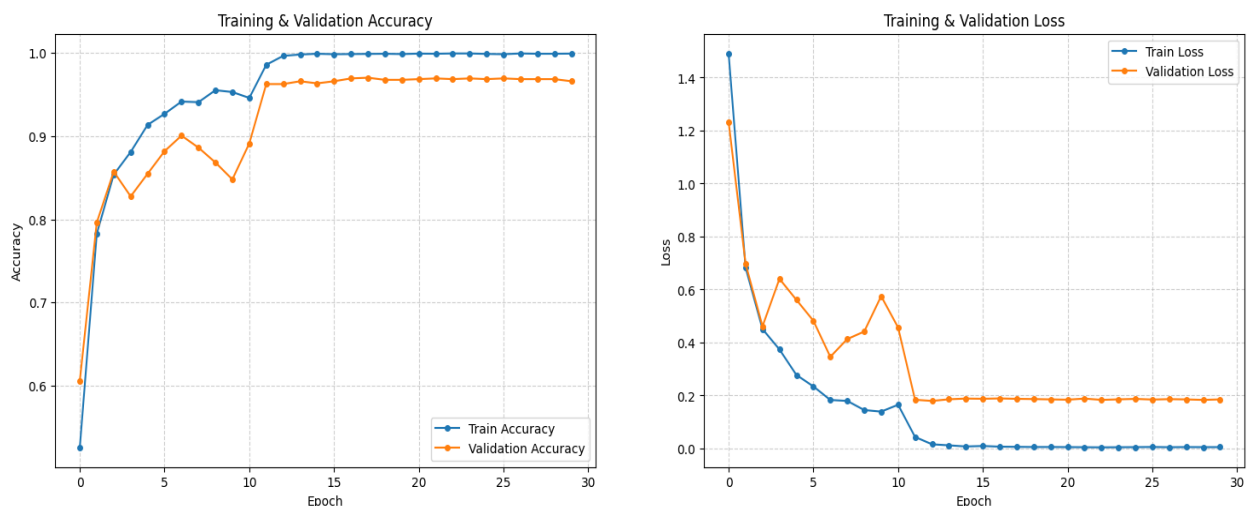
Tabel 3. Perbandingan skenario prapemrosesan dan augmentasi pada data latih, validasi, dan uji

Tahapan Preprocessing data	Data Latih	Data validasi dan Uji
Resize (224 x 224)	√	√
Random Rotation (15°)	√	-
Random Horizontal Flip	√	-
To Tensor	√	√
Normalize	√	√

3.3 Implementasi model

Model klasifikasi Bahasa Isyarat Alfabet Arab diimplementasikan menggunakan arsitektur Deep Learning ResNet-50. Pengembangan dilakukan di atas framework PyTorch, yang dipilih karena fleksibilitasnya dalam komputasi tensor dan penyusunan grafik dinamis. Sesuai dengan metodologi, strategi Transfer Learning diterapkan dengan memuat bobot pra-latih (pre-trained weights) dari ImageNet. Lapisan klasifikasi akhir (Fully Connected Layer) dimodifikasi untuk memetakan fitur ke dalam 31 kelas target alfabet Arab. Pelatihan dijalankan menggunakan optimizer Adam dengan learning rate awal 0.001 dan fungsi kerugian Cross Entropy Loss. Selain itu, mekanisme Learning Rate Scheduler (ReduceLROnPlateau) diterapkan guna mencapai konvergensi yang lebih stabil dan optimal.

Proses pelatihan model resnet50 dilakukan selama 30 epoch dengan melihat dua metrix utama yaitu accuracy dan loss pada data training dan validation. Grafik hasil pelatihan model di setiap epoch ditunjukkan pada Gambar 7 berikut.



Gambar 7. Kurva metrik *accuracy* dan *loss* selama proses pelatihan dan validasi model ResNet-50.

Merujuk pada kurva pelatihan pada Gambar 7, puncak performa (*peak performance*) model tercapai pada *epoch* ke-18. Pencapaian optimal pada titik ini mengindikasikan bahwa model telah mencapai konvergensi yang ideal untuk mengenali fitur kompleks dari data latih. Pada *epoch* tersebut, model mencatatkan hasil evaluasi terbaik dengan Validasi Akurasi (*Val Acc*) mencapai 97,03%, Validasi Loss (*Val Loss*) di angka 0,1875, dan Akurasi Pelatihan (*Train Acc*) sebesar 99,89%. Selisih (*gap*) yang sangat sempit antara akurasi pelatihan dan validasi (sekitar 2,8%) membuktikan bahwa model berada dalam kondisi *good fit* dan terhindar dari *overfitting*.

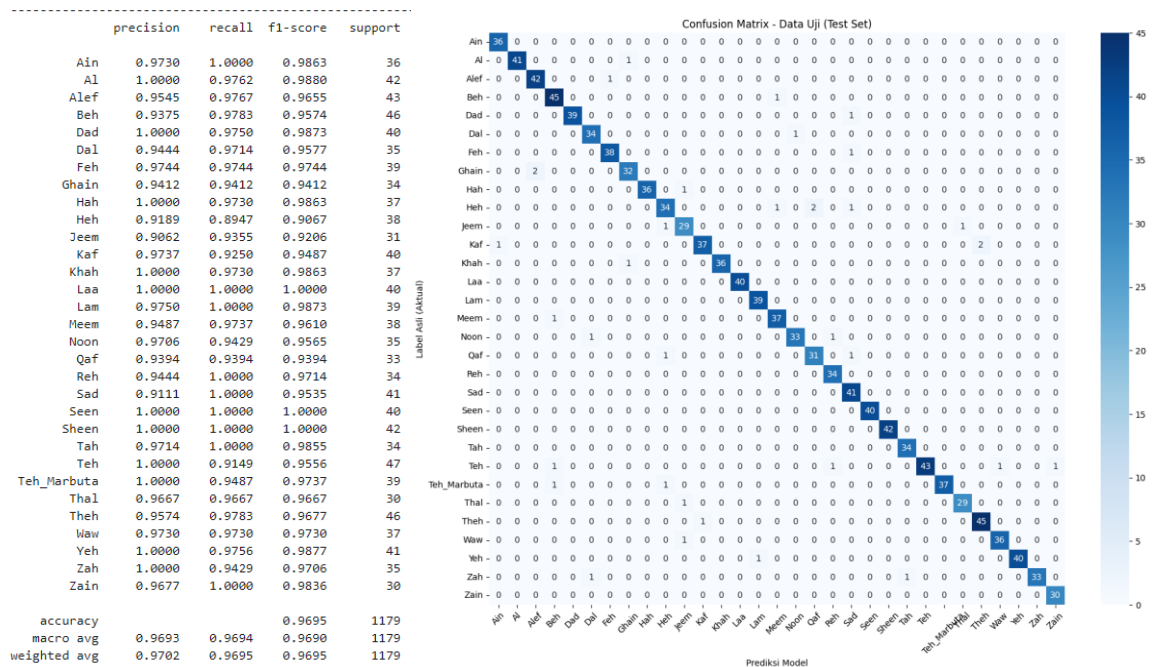
Stabilitas pencapaian performa ini didukung oleh implementasi penjadwalan laju pembelajaran (*Learning Rate Scheduler*) yang secara otomatis menurunkan nilai *learning rate* ketika metrik evaluasi mulai stagnan, sehingga memfasilitasi model untuk mencapai titik konvergensi yang lebih halus. Lebih lanjut, untuk memastikan performa puncak ini tidak hilang pada sisa iterasi berikutnya, penelitian ini mengimplementasikan mekanisme *Model Checkpoint* guna menyimpan bobot jaringan (*weights*) terbaik secara otomatis tepat pada *epoch* ke-18 tersebut.

3.4 Pengujian model

Evaluasi kinerja model dilakukan melalui serangkaian skenario pengujian komprehensif untuk mengukur efektivitas arsitektur ResNet-50 dalam mengenali klasifikasi Bahasa Isyarat Alfabet Arab. Pengujian dirancang pada dua kondisi data yang kontras, yaitu kondisi ideal (citra bersih) dan kondisi simulasi bawah air (*underwater simulation*). Parameter evaluasi yang digunakan meliputi pengukuran Accuracy, Precision, Recall, F1-Score, serta analisis kesalahan prediksi menggunakan Confusion Matrix.

a. Pengujian skenario 1 (Kondisi ideal)

Pengujian pertama dilakukan menggunakan data uji (*test set*) yang merepresentasikan kondisi ideal (pencahayaan standar, tanpa *noise*). Pada skenario ini, model ResNet-50 mencapai akurasi sebesar 96,95%. Hasil ini membuktikan bahwa metode *transfer learning* efektif untuk mengenali pola gestur tangan alfabet Arab ketika kualitas citra masukan optimal, Dimana ditunjukkan oleh data evaluasi pada Gambar 8 berikut.

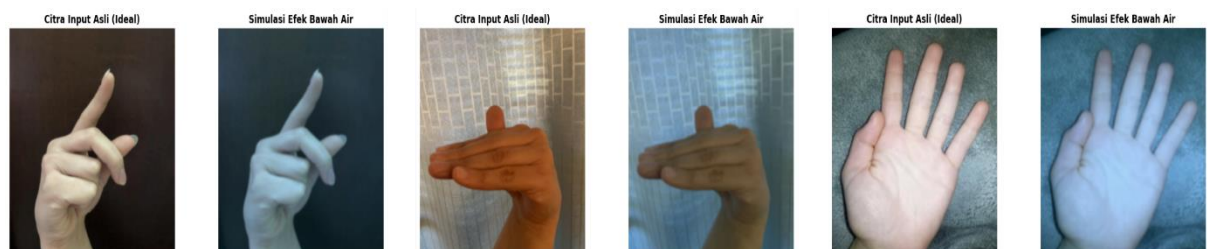


Gambar 8. Data evaluasi dan confusion matrix hasil klasifikasi pada data ideal.

Dari visualisasi *Confusion Matrix* pada gambar 8 di atas, terlihat bahwa mayoritas prediksi berkumpul di garis diagonal utama, yang mengindikasikan ketepatan klasifikasi yang tinggi di hampir seluruh kelas huruf.

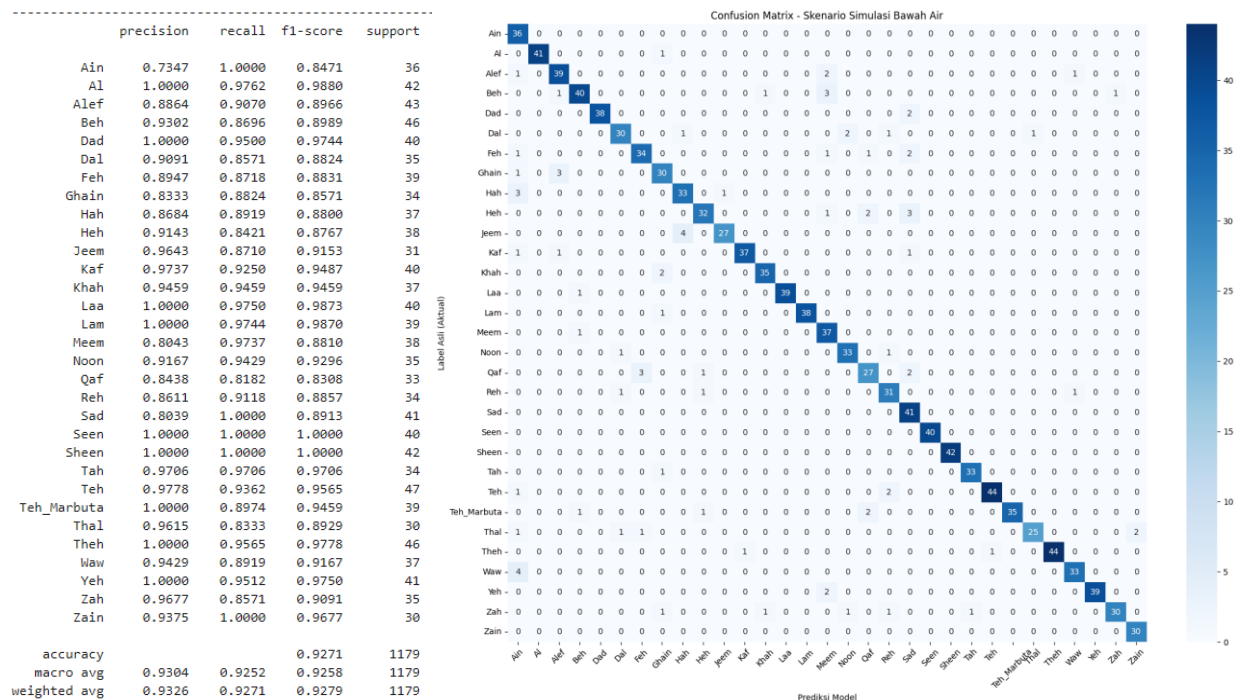
b. Pengujian skenario 2 (Kondisi simulasi bawah air)

Untuk menjawab rumusan masalah mengenai ketahanan arsitektur, dilakukan pengujian simulasi kondisi bawah air (*underwater*). Penelitian ini menerapkan teknik manipulasi citra digital (*Digital Image Processing*) pada data uji untuk meniru karakteristik visual air. Efek simulasi dibangun melalui serangkaian filter bertahap yang mencakup manipulasi kanal warna dengan penyesuaian intensitas RGB untuk meniru penyerapan cahaya merah di air, penerapan lapisan warna (*tint*) biru laut menggunakan teknik *overlay* dan *blending* dengan transparansi sebesar 0.2, serta penambahan *Gaussian Blur* untuk mensimulasikan kekeruhan air dan pembiasan cahaya. Perbandingan visual antara citra input ideal dan citra hasil simulasi dapat dilihat pada Gambar 9 berikut:



Gambar 9. Perbandingan visual antara citra input ideal dan simulasi bawah air.

Setelah dilakukan pengujian pada dataset yang telah terdistorsi, didapatkan hasil pengukuran data evaluasi yang di tunjukkan pada gambar 10. berikut.



Gambar 10. Data evaluasi dan confusion matrix hasil klasifikasi pada data simulasi bawah air.

Pada matriks di Gambar 10. terlihat adanya peningkatan penyebaran prediksi di luar diagonal utama (mis-klasifikasi) dibandingkan kondisi ideal. Hal ini mengindikasikan bahwa distorsi visual bawah air menyebabkan model kesulitan membedakan huruf-huruf dengan kemiripan gestur. Degradasi performa paling menonjol terlihat pada kelas 'Ain' yang memiliki tingkat kesalahan positif (false positive) cukup tinggi, serta penurunan sensitivitas pada huruf 'Qaf'. Tingginya tingkat kesalahan pada kedua kelas ini secara teknis disebabkan oleh karakteristik gesturnya yang sangat bergantung pada fitur spasial halus (*fine-grained features*), seperti posisi tumpang tindih antar jari, tekukan sendi, atau penempatan spesifik ibu jari. Distorsi bawah air, khususnya efek kekeruhan (*blur*) dan penurunan kontras, menghilangkan detail tepi (*edge details*) berfrekuensi tinggi tersebut, sehingga bentuk jari tampak menyatu dan model kehilangan fitur diskriminatif utamanya. Meskipun demikian, model tetap menunjukkan ketahanan (*robustness*) yang baik pada huruf dengan fitur geometri makro yang tegas dan siluet terbuka, seperti rentangan lima jari pada isyarat 'Seen' dan 'Sheen' yang strukturnya tidak mudah rusak oleh degradasi visual.

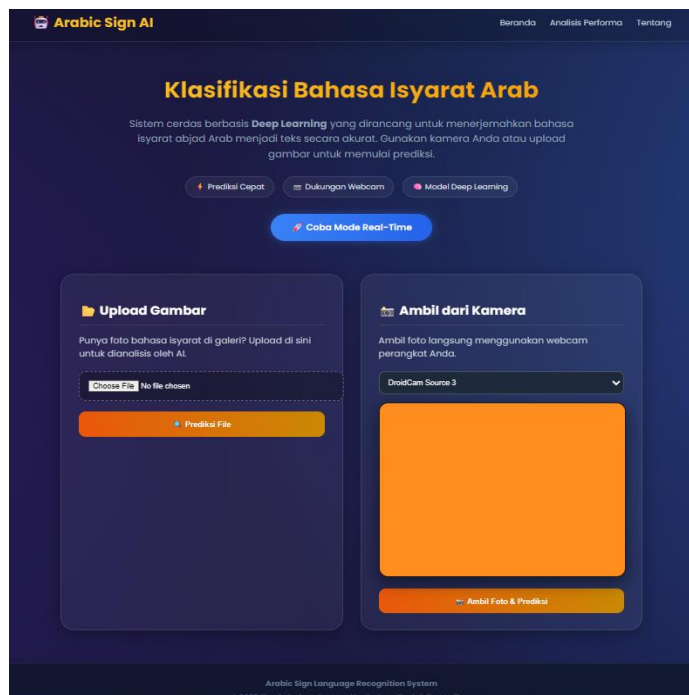
Tabel 4. Adalah hasil komparasi pengujian kondisi ideal dan terdistorsi

Pengujian	Accuracy	Precision	Recall	F1-Score
Kondisi ideal	0.9695	0.9702	0.9695	0.9695
Kondisi terdistorsi	0.9271	0.9326	0.9271	0.9279
Penurunan	4.24%	3.76%	4.24%	4.16%

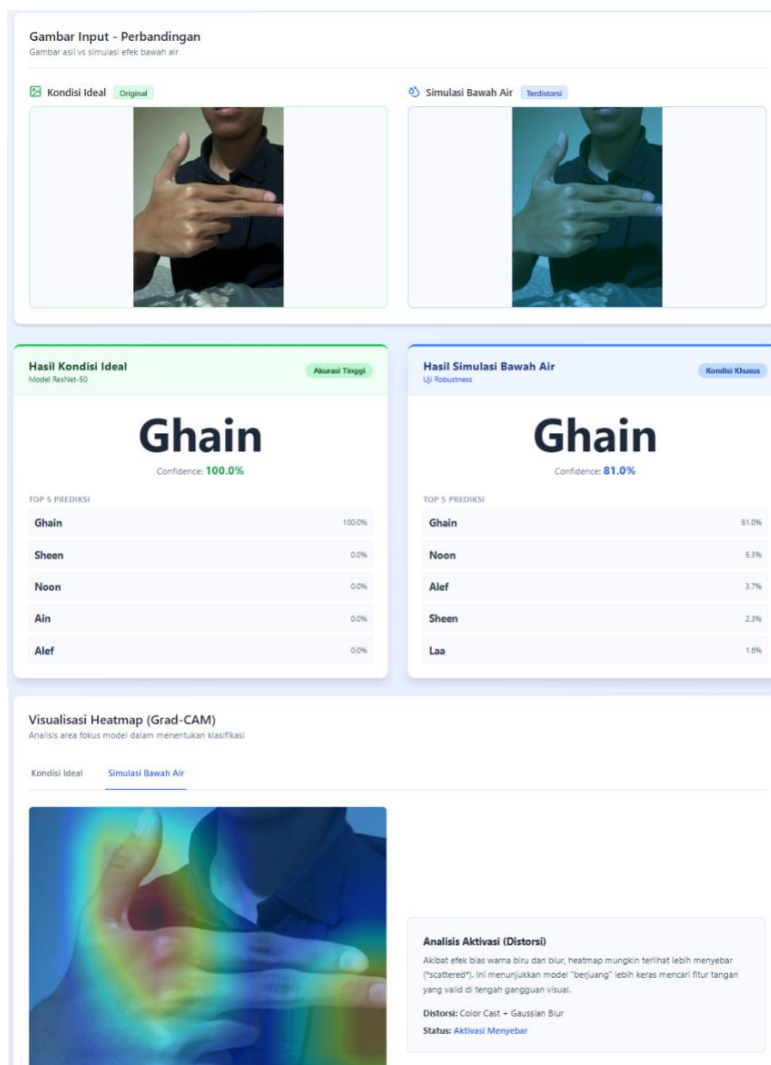
Hasil komparasi pengujian yang ditunjukkan pada Tabel 6 model klasifikasi menunjukkan performa yang sangat baik pada kondisi ideal dengan tingkat akurasi mencapai 0.9695 (96.95%). Namun, ketika dihadapkan pada citra dengan kondisi terdistorsi, model mengalami penurunan kinerja di seluruh metrik evaluasi. Penurunan terbesar terjadi pada metrik akurasi dan *recall*, yakni sebesar 4.24%, yang mengindikasikan bahwa distorsi visual cukup memengaruhi kemampuan model dalam mengidentifikasi fitur target secara tepat. Meskipun demikian, F1-Score pada kondisi terdistorsi masih berada di angka 0.9279, menunjukkan bahwa model arsitektur yang digunakan masih memiliki ketahanan (*robustness*) yang tergolong baik dalam menangani anomali visual pada dataset.

3.5 Visualisasi implementasi model

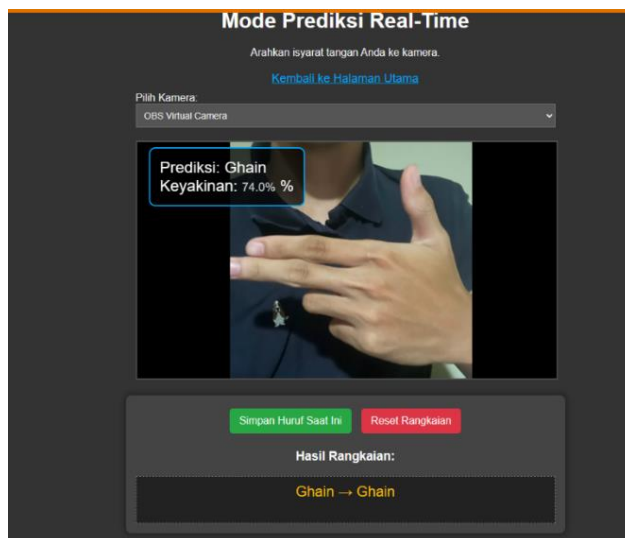
Model yang telah dilatih kemudian diimplementasikan ke dalam sebuah prototipe aplikasi web interaktif sebagai bentuk validasi operasional. Fungsionalitas utama dari aplikasi berbasis web ini mencakup klasifikasi citra statis melalui unggahan berkas (*upload*) gambar, pemrosesan foto yang ditangkap secara langsung melalui antarmuka situs, serta prediksi gestur secara *real-time* yang memanfaatkan aliran video dari kamera pengguna. Secara visual, antarmuka utama serta representasi luaran prediksi dari prototipe aplikasi web interaktif tersebut ditunjukkan secara berurutan pada Gambar 11, 12, dan 13 berikut.



Gambar 11. Tampilan dashboard website visualisasi model prediksi isyarat alfabet arab



Gambar 12. Tampilan hasil prediksi dari pengenalan isyarat alfabet arab yang berisi heatmap dan perbandingan hasil prediksi dari kondisi ideal serta kondisi simulasi bawah air.



Gambar 13. Adalah tampilan hasil prediksi isyarat alfabet arab secara real time.

3.6 Pembahasan

Berbeda dengan literatur terdahulu yang umumnya berfokus mengejar akurasi absolut, penelitian ini menitikberatkan pada evaluasi ketahanan (*robustness*) model. Tinjauan oleh Alayed (2024) [3] menunjukkan model *State-of-the-Art* (SOTA) mampu mencapai akurasi 100% pada kondisi visual terkontrol. Sementara itu, studi Brianorman dan Munir (2023) [2] yang sama-sama menggunakan *transfer learning* mencatatkan *underperformance* pada arsitektur ResNet-50 dengan akurasi pengujian yang hanya mencapai 82,58%. Sebaliknya, strategi optimasi dalam penelitian ini berhasil mencatatkan kinerja *baseline* ResNet-50 yang jauh lebih superior, yakni 96,95% pada kondisi ideal. Lebih lanjut, untuk merespons kerentanan arsitektur terhadap gangguan optik [8], pengujian simulasi bawah air membuktikan bahwa model tetap tangguh meski akurasi turun 4,24% menjadi 92,71%. Komparasi ini secara empiris mengonfirmasi bahwa model yang dikembangkan tidak hanya melampaui performa literatur sebelumnya secara *head-to-head* pada kondisi ideal, tetapi juga menawarkan pemetaan batas toleransi (*boundary limits*) yang realistis dengan mempertahankan *F1-Score* di angka 0,9279 di tengah paparan distorsi visual ekstrem.

4. KESIMPULAN

Penelitian ini mengonfirmasi bahwa arsitektur ResNet-50 memiliki kinerja superior dalam mengklasifikasikan Bahasa Isyarat Alfabet Arab dengan akurasi 96,95% pada kondisi ideal, namun tetap mengalami degradasi saat dihadapkan pada distorsi visual ekstrem. Paparan simulasi bawah air terbukti memicu penurunan akurasi sebesar 4,24%, yang secara teknis mengindikasikan bahwa anomali visual memengaruhi efektivitas ekstraksi fitur geometri tangan. Meskipun demikian, keberadaan *residual connections* pada model terbukti mampu memberikan margin ketahanan yang baik sehingga *F1-Score* tetap dapat dipertahankan di angka 0,9279. Terlepas dari reduksi performa tersebut, keandalan operasional model telah berhasil divalidasi secara *real-time* melalui prototipe aplikasi web interaktif berbasis Flask dan SocketIO. Mengingat evaluasi ini masih terbatas pada efek bawah air, penelitian selanjutnya disarankan untuk mengeksplorasi skenario distorsi multi-kondisi guna memetakan profil ketahanan sistem cerdas secara menyeluruh di berbagai lingkungan *non-ideal*.

REFERENCES

- [1] I. Bagus, A. Peling, I. M. Pasek, A. Ariawan, and G. B. Subiksa, "Deteksi Bahasa Isyarat Menggunakan Tensorflow Lite dan American Sign Language (ASL)," *J. Krisnadana*, vol. 3, no. 2, pp. 90–100, 2024, [Online]. Available: <https://doi.org/10.58982/krisnadana.v3i2.534>
- [2] Y. Brianorman and R. Munir, "Perbandingan Pre-Trained CNN : Klasifikasi Pengenalan Bahasa Isyarat Huruf Hijaiyah," *J. Sist. Inf. Bisnis*, vol. 13, no. 1, pp. 52–59, 2023, doi: 10.21456/vol13iss1pp52-59.
- [3] A. Alayed, "Machine Learning and Deep Learning Approaches for Arabic Sign Language Recognition: A Decade Systematic Literature Review.," *Sensors*, vol. 24, no. 23, p. 7798, 2024, [Online]. Available: <https://doi.org/10.3390/s24237798>
- [4] A. Rakhmadi, A. Yudhana, and S. Sunardi, "A Study Of Worldwide Patterns In Alphabet Sign Language Recognition Using Convolutional And Recurrent Neural Networks," *J. Tek. Inform.*, vol. 6, no. 1 SE-Articles, pp. 187–204, Feb. 2025, doi: 10.52436/1.jutif.2025.6.1.4202.
- [5] T. P. Munthe and M. Akbar, "Klasifikasi Citra Biji Kopi Temanggung Menggunakan Residual Network," *J. Pustaka Data*, vol. 5, no. 1, pp. 94–102, 2025, [Online]. Available: <https://doi.org/10.30591/smartcomp.v1i1i2.3527>
- [6] H. Imaduddin, I. C. Utomo, and D. A. Anggoro, "Fine-tuning ResNet-50 for the classification of visual impairments from retinal fundus images," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 4, pp. 4175–4182, 2024, doi: 10.11591/ijece.v14i4.pp4175-4182.



- [7] A. . S. Rakhmadi, A.; Yudhana, “Integrating Social Responsibility in the Development of Sign Language Recognition Technology for Quranic Recitation,” in *2025 International Conference on Information Technology and Computing (ICITCOM)*, Yogyakarta: IEEE, 2025, pp. 30–35. doi: 10.1109/ICITCOM66635.2025.11265077.
- [8] L. Wang, S. Chen, and F. Chien, “Robust deep convolutional neural network against image distortions,” *APSIPA Trans. Signal Inf. Process.*, vol. 10, no. 1, p. e14, 2021, doi: 10.1017/ATSIP.2021.14.
- [9] N. A. Putra and A. K. Wardhana, “Image-Based Classification of Healthy and Unhealthy Goats Using ResNet-18 Deep Learning Model,” *J. Appl. Informatics Comput.*, vol. 9, no. 5, pp. 2357–2363, 2025, [Online]. Available: <https://doi.org/10.30871/jaic.v9i5.10267>
- [10] A. . S. Rakhmadi, A.; Khairunnisa, S.; Adhantoro, M. S.; Riyadi, S.; Yudhana, “EfficientNetB0-Based Recognition of Arabic Sign Language for Quranic Recitation Support,” in *2025 12th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Semarang: IEEE, 2025, pp. 137–143. doi: 10.1109/EECSI67060.2025.11290662.
- [11] Suharyanto, Frieyadi, and S. J. Kuryanti, “Peningkatan Kualitas Citra Bawah Air Berbasis Algoritma Fusion Dengan Keseimbangan Warna, Optimalisasi Kontras, Dan Peregangkan Histogram,” *INTI Nusa Mandiri*, vol. 16, no. 1, pp. 31–38, 2021, [Online]. Available: <https://doi.org/10.33480/inti.v16i1.2286>
- [12] H. Imaduddin and B. A. Hermansyah, “Transfer learning for detecting COVID-19 on x-ray using deep residual network,” *Bull. Electr. Eng. Informatics*, vol. 11, no. 6, pp. 3414–3421, 2022, doi: 10.11591/eei.v11i6.4334.
- [13] R. F. M. Bang and A. Y. Chandra, “Analisis Perbandingan Kinerja Model CNN Resnet-50 , VGG19 dan Mobilenet dalam Klasifikasi Penyakit pada Tanaman Mete,” *J. LOCUS Penelit. Pengabd.*, vol. 4, no. 8, pp. 7903–7918, 2025, [Online]. Available: <https://doi.org/10.58344/locus.v4i8.4261>
- [14] D. Banu and D. Hanggoro, “Analisis Komparatif Arsitektur Deep Learning Untuk Aplikasi Computer Vision : Studi Literature Review,” *J. Komput. Teknol. Inf. Sist. Inf.*, vol. 4, no. 2, pp. 1001–1008, 2025, [Online]. Available: <https://doi.org/10.62712/juktisi.v4i2.542>
- [15] F. Nashrullah, S. Adhi, and G. Budiman, “Investigasi Parameter Epoch Pada Arsitektur ResNet- 50 Untuk Klasifikasi Pornografi,” *J. Comput. Electron. Telecommun.*, vol. 1, no. 1, p. 1:8, 2020, [Online]. Available: <https://doi.org/10.52435/complete.v1i1.51>
- [16] S. Aras and A. Setyanto, “Deep Learning Untuk Klasifikasi Motif Batik Papua Menggunakan EfficientNet dan Transfer Learning,” *Insect (Informatics Secur. J. Tek. Inform.*, vol. 8, no. 1, pp. 11–20, 2022, [Online]. Available: <https://doi.org/10.33506/insect.v8i1.1865>
- [17] H. Imaduddin, F. Y. A’la, A. Fatmawati, and B. A. Hermansyah, “Comparison of transfer learning method for COVID-19 detection using convolution neural network,” *Bull. Electr. Eng. Informatics*, vol. 11, no. 2, pp. 1091–1099, 2022, doi: 10.11591/eei.v11i2.3525.
- [18] A. Rakhmadi, A. Yudhana, and S. Sunardi, “CNN-Based SIBI Sign Language Recognition Alphabet: Exploring the Impact of Hardware on Model Training,” *J. Appl. Eng. Technol. Sci.*, vol. 7, no. 1 SE-Articles, pp. 224–246, Dec. 2025, doi: 10.37385/jaets.v7i1.7071.
- [19] M. T. Rustam, E. Syahrin, and A. Syamanta, “Membangun Komputer Vision Deteksi Jari dengan Menampilkan Angka dengan Pytorch,” *Indones. J. Interdiscip. Res. Sci. Technol.*, vol. 3, no. 5, pp. 521–528, 2025, [Online]. Available: <https://doi.org/10.55927/marcopolo.v3i5.43>
- [20] A. Rakhmadi, A. Yudhana, Sunardi, and S. Riyadi, “Enhancing Arabic Sign Recognition with ResNet: Towards Inclusive Quranic Technologies,” in *2025 12th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, IEEE, 2025, pp. 1–7. doi: 10.1109/ICITACEE66165.2025.11232873.
- [21] A. Rakhmadi, A. Yudhana, and S. Sunardi, “VGG16-Based Feature Extraction for Arabic Alphabet Sign Language Classification to Support Qur’anic Tadarus Accessibility,” *J. Tek. Inform.*, vol. 6, no. 4 SE-Articles, pp. 2602–2624, Aug. 2025, doi: 10.52436/1.jutif.2025.6.4.4953.
- [22] D. F. Ningtyas and N. Setiyawati, “Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request,” *J. Janitra Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 19–34, 2021, doi: 10.25008/janitra.v1i1.120.
- [23] E. H. Y. Kristianto and N. Setiyawati, “Pembangunan Aplikasi virtual Inventori System (VIS) Berbasis Web Menggunakan Flask Framework (Studi Kasus: PT XYZ),” *J. Mnemon.*, vol. 4, no. 2, pp. 128–135, 2021, [Online]. Available: <https://doi.org/10.36040/mnemonic.v5i2.4799>
- [24] S. J. Karam and B. F. Abdulrahman, “Using Socket . io Approach for Many-to-Many Bi-Directional Video Conferencing,” *Al-Rafidain J. Comput. Sci. Math.*, vol. 16, no. 1, pp. 81–86, 2022, [Online]. Available: <https://doi.org/10.33899/csmj.2022.174411%0A>