

# KLASIFIKASI PENGENALAN HURUF HIJAIYAH PADA BAHASA ISYARAT ARAB MENGGUNAKAN *TRANSFER LEARNING EFFICIENTNETB1*

Diani Eka Putri<sup>\*1)</sup>, Jumadi<sup>2)</sup>, Eva Nurlatifah<sup>3)</sup>

1. Teknik Informatika, Sains dan Teknologi, Universitas Islam Negeri Sunan Gunung Djati, Indonesia
2. Teknik Informatika, Sains dan Teknologi, Universitas Islam Negeri Sunan Gunung Djati, Indonesia
3. Teknik Informatika, Sains dan Teknologi, Universitas Islam Negeri Sunan Gunung Djati, Indonesia

## Article Info

**Kata Kunci:** Bahasa Isyarat Arab; *Convolutional Neural Network*; *EfficientNetB1*; *Fine-Tuning*; *Transfer Learning*

**Keywords:** *Arabic Sign Language*; *Convolutional Neural Network*; *EfficientNetB1*; *Fine-Tuning*; *Transfer Learning*

## Article history:

Received 11 November 2024

Revised 15 Desember 2024

Accepted 14 Januari 2025

Available online 15 Maret 2025

## DOI :

<https://doi.org/10.29100/jipi.v10i2.6174>

\* Corresponding author.

Corresponding Author

E-mail address:

[daniekaputri28@gmail.com](mailto:daniekaputri28@gmail.com)

## ABSTRAK

Individu Muslim dengan gangguan pendengaran sering menghadapi tantangan dalam mempelajari Al-Qur'an, terutama karena kurangnya sistem pendukung untuk pembelajaran bahasa isyarat Arab. Bahasa isyarat memainkan peran penting dalam memfasilitasi komunikasi yang efektif bagi mereka yang mengalami kesulitan pendengaran. Untuk mengatasi masalah ini, penelitian ini bertujuan untuk mengembangkan sistem pengenalan Bahasa Isyarat Arab (ArSL) dengan menggunakan *Convolutional Neural Network* (CNN) dan pendekatan *Transfer Learning* menggunakan model *pre-trained EfficientNetB1*. Sistem ini dirancang untuk mengidentifikasi gerakan bahasa isyarat Arab dari *dataset* yang terdiri dari 28 huruf Hijaiyah, dengan masing-masing huruf memiliki 100 citra. Pengujian dilakukan pada model dengan empat skenario berbeda untuk menemukan konfigurasi yang paling optimal. Hasil penelitian menunjukkan bahwa dengan membekukan 30% lapisan awal model selama *fine-tuning* menghasilkan akurasi 98.95% pada data *train* dan 99.52% pada data *validation*. Pada data *test*, model mencapai performa terbaik dengan *accuracy*, *precision*, *recall*, dan *f1-score* masing-masing sebesar 100%.

## ABSTRACT

*Muslim individuals with hearing loss often face challenges in learning the Qur'an, mainly due to the lack of support systems for Arabic sign language learning. Sign language plays an important role in facilitating effective communication for those with hearing difficulties. To address this issue, this research aims to develop an Arabic Sign Language (ArSL) recognition system by using Convolutional Neural Network (CNN) and Transfer Learning approach using a pre-trained EfficientNetB1 model. The system is designed to identify Arabic sign language gestures from a dataset consisting of 28 Hijaiyah letters, with each letter having 100 images. Tests were conducted on the model with four different scenarios to find the most optimal configuration. The results showed that freezing 30% of the initial layer of the model during fine-tuning resulted in an accuracy of 98.95% on train data and 99.52% on validation data. On test data, the model achieved the best performance with accuracy, precision, recall, and f1-score of 100% each.*

## I. PENDAHULUAN

**B**ERDASARKAN data statistik sekolah luar biasa tahun 2018/2019 mencatat bahwa terdapat 26.438 jiwa penyandang tunarungu dari total penduduk Indonesia yang berjumlah 266,91 juta jiwa pada pertengahan tahun 2019[1]. Individu yang mengalami gangguan pendengaran memiliki kemampuan mendengar yang terbatas, dan hal ini tidak sepenuhnya dapat diatasi dengan alat bantu dengar. Terdapat keterbatasan dalam memenuhi berbagai kebutuhan penyandang disabilitas tunarungu secara optimal, terutama dalam aspek sarana, prasarana, dan sumber daya manusia. Oleh karena itu, langkah signifikan yang perlu diambil adalah memberikan edukasi mengenai bahasa isyarat sebagai identitas bagi penyandang tunarungu [2].

Menurut penelitian yang dilakukan oleh Rahmat et al. (2021), anak-anak yang mengalami hambatan pendengaran sering menghadapi kesulitan dalam membaca dan mempelajari Al-Quran. Hal tersebut dikarenakan kurangnya metode pengajaran yang memadai untuk memenuhi kebutuhan pembelajaran anak-anak dengan hambatan

pendengaran [3]. Bahasa isyarat menjadi elemen penting dalam komunikasi penyandang tunarungu sebagai sarana interaksi. Dalam bahasa isyarat, terdapat beragam tanda yang dibedakan oleh variasi bentuk tangan, posisi tangan, gerakan, ekspresi wajah, dan bagian tubuh lainnya untuk menyampaikan makna tertentu termasuk dalam konteks pembelajaran agama bagi mereka di Indonesia.

Dalam Pasal 14 ayat c tentang Hak Keagamaan Undang-Undang Penyandang Disabilitas Nomor 8 Tahun 2016, disebutkan bahwa penyandang disabilitas berhak mendapatkan kitab suci dan lektur keagamaan lainnya yang mudah diakses berdasarkan kebutuhannya. Implementasi sila pertama menyatakan bahwa pembelajaran agama menjadi kewajiban di setiap sekolah umum termasuk sekolah luar biasa. Salah satunya adalah pembelajaran agama Islam. Aspek penting dalam pembelajaran agama Islam adalah mempelajari Al-Quran. Tahap pertama yang dilakukan untuk mempelajari Al-Quran dimulai dengan pengenalan huruf Hijaiyah. Kewajiban dalam membaca Al-Quran dianggap sebagai pedoman hidup, sesuai dengan berbagai ayat yang tertulis di dalam Al-Quran salah satunya adalah firman Allah Ta'ala pada QS. Al-Alaq (96) ayat 1-5 :

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ۝١ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ۝٢ اقْرَأْ وَرَبُّكَ الْأَكْرَمُ ۝٣ الَّذِي عَلَّمَ بِالْقَلَمِ ۝٤ عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ ۝٥

Artinya:

“1) Bacalah dengan (menyebut) nama Tuhanmu yang menciptakan. 2) Dia telah menciptakan manusia dari segumpal darah. 3) Bacalah, dan Tuhanmulah yang Maha Mulia. 4) Yang mengajarkan (manusia) dengan perantara kalam. 5) Dia mengajarkan manusia apa yang tidak diketahuinya”.

Dengan demikian, penerapan teknologi dalam pengenalan bahasa isyarat, khususnya dalam pembelajaran Al-Quran, sangat penting untuk meningkatkan aksesibilitas dan efektivitas pembelajaran bagi penyandang disabilitas.

Saat ini, teknologi *Deep Learning* diterapkan secara luas dalam pengembangan bahasa isyarat untuk meningkatkan kualitas hidup penyandang disabilitas. Sejumlah penelitian telah dilakukan untuk memperbaiki sistem pengenalan bahasa isyarat dalam berbagai bahasa dengan menerapkan berbagai teknik. Bani et al. (2023) mengembangkan sistem pengenalan Bahasa Isyarat Arab menggunakan *dataset* berisi 54.049 gambar huruf *ArSL*. Penelitian ini mengimplementasikan enam arsitektur *pre-trained*, termasuk *MobileNetV2*, *VGG16*, *InceptionV3*, *ResNet50V2*, *ResNet152*, dan *Xception*. Model *InceptionV3* mencapai akurasi 100% dan *loss score* 0.00 tanpa mengalami *overfitting* [4]. Duwairi dan Halloush (2022) mengusulkan kerangka kerja berbasis model *machine learning* seperti *AlexNet*, *VGGNet*, dan *GoogLeNet/Inception* untuk pengenalan otomatis Bahasa Isyarat Arab. Model *VGGNet* mencapai tingkat akurasi 97% [5]. AlKhuraym et al. (2023) menggunakan model *EfficientNet* yang dimodifikasi untuk mengkategorikan gerakan Bahasa Isyarat Arab. Penelitian ini mengembangkan kumpulan data gerakan tangan untuk 30 huruf Arab berbeda dan memperoleh tingkat akurasi 94% dengan *EfficientNet-Lite 0* [6]. Hayani et al. (2019) mengusulkan pendekatan baru berdasarkan *Convolutional Neural Network* (CNN) untuk mengenali angka dan huruf dalam Bahasa Isyarat Arab. Sistem yang diusulkan mencapai tingkat pengenalan 90,02%, melampaui *Support Vector Machine* (SVM) dengan 88% dan *K-nearest neighbour* (KNN) dengan 66% [7]. Dengan menggunakan arsitektur *EfficientNet-B0*, Sundari et al. (2023) mengklasifikasikan jenis kulit kering, normal, dan kombinasi. Hasil pengujian menunjukkan bahwa sistem mencapai *accuracy* 100%, *loss* 0,001, *precision* 100%, *recall* 100%, dan *f1-score* 100% [8]. Wardana et al. (2021) menggunakan arsitektur *EfficientNet B4* untuk membuat sistem pengenalan bahasa isyarat. *Dataset Alphabets ASL* terdiri dari 87.000 gambar dari 29 kelas. Untuk mengurangi kompleksitas, metode *Cyclical Learning Rate (CLR)* dan ukuran input 64 x 64 x 3 digunakan. Hasil evaluasi menunjukkan bahwa data tes memiliki akurasi 99,81% dan waktu pengenalan gestur adalah 0,056 detik [9]. Perdani et al. (2022) menggunakan arsitektur *EfficientNet* untuk membuat sistem klasifikasi tingkatan *glaukoma* menggunakan gambar *fundus* mata. Data gambar berasal dari *dataset RimOne r1*, dan klasifikasi dibagi menjadi 5 kelas yaitu *early*, *moderate*, *deep*, *OHT*, dan normal. Model yang paling efektif menggunakan *optimizer Adamax*, *learning rate* 0,001, *epoch* 20, dan ukuran *batch* 32. Hasil tes menunjukkan *f1-score*, *accuracy*, *precision*, dan *recall* masing-masing 1,0000 [10]. Al Rivan dan Hartoyo (2022) mengklasifikasikan bahasa isyarat Indonesia menggunakan *CNN* dengan *VGG-16* dan *AlexNet*. Hasilnya, *VGG-16* dengan *optimizer Adam* mencapai akurasi tertinggi 99,32% per huruf dan 91,18% secara keseluruhan, sedangkan *AlexNet* dengan *Adam* mencapai 99,16% per huruf dan 89,04% secara keseluruhan [11]. Brianorman dan Munir (2023) melakukan penelitian menggunakan empat model *CNN pre-trained* untuk pengenalan huruf hijaiyah berbasis gambar dengan jumlah 6.200 *dataset*. *VGG16*, *Xception*, dan *MnetV2* mencapai akurasi 99,85% masing-masing dalam 2, 3, dan 11 epoch, sementara *ResNet50* hanya mencapai 82,12% setelah 100 epoch. Pada pengujian, *VGG16*, *Xception*, dan *MnetV2* memperoleh *precision*, *recall*, *f1-score*, dan *accuracy* sebesar 100%. *ResNet50* mencapai 81,55% *precision*, 86,04% *recall*, 82,04% *f1-score*, dan 82,58% *accuracy*. Implementasi *MnetV2* menunjukkan kinerja yang baik untuk pengenalan bentuk jari secara *real time* [12]. Penelitian Arisandi dan Satya

(2022) menggunakan teknologi komputer untuk mengidentifikasi bahasa isyarat sebagai solusi untuk komunikasi antara tuna rungu atau bisu dan orang normal. Sistem penerjemah bahasa otomatis ini memproses gambar isyarat huruf A hingga E, I, You, dan I Love You menggunakan arsitektur *Convolutional Neural Network (CNN)*. Dengan metode ini, nilai akurasi yang dicapai bisa mencapai 99,82% [13].

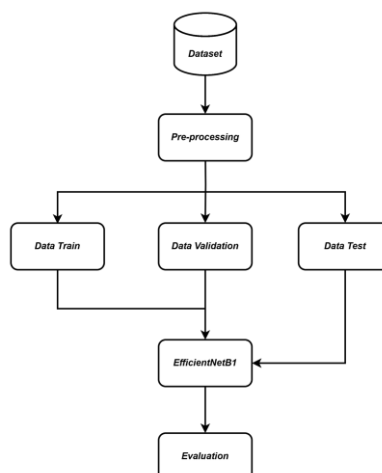
Teknologi *Deep Learning* telah membuat kemajuan signifikan dalam pengenalan bahasa isyarat. Model seperti *MobileNet*, *VGGNet*, *Inception*, *ResNet* dan *Xception* membantu meningkatkan efisiensi dan akurasi sistem. *MobileNet* terkenal karena efisiensinya dalam ukuran dan kecepatan inferensi. *VGGNet* dengan arsitektur yang rumit memiliki kemampuan untuk mengekstrak fitur visual secara detail. Namun, seringkali diperlukan sumber daya komputasi yang lebih besar. *Inception* unggul dalam menangkap fitur multi-skala dengan menggunakan berbagai ukuran filter. Sementara *ResNet* dan *Xception* menggunakan teknik residual dan secara *depthwise separable convolutions* untuk menangkap fitur mendalam secara efektif. Di antara model-model ini, *EfficientNetB1* adalah model yang akan digunakan pada penelitian ini karena memiliki kemampuan untuk menyeimbangkan akurasi tinggi dengan efisiensi komputasi yang optimal melalui pendekatan *compound scaling*.

Namun, penelitian sebelumnya menghadapi berbagai tantangan, seperti risiko *overfitting* pada model yang sangat kompleks. Selain itu, beberapa sistem tertentu membutuhkan sumber daya komputasi yang tinggi, yang membatasi aplikasi pada perangkat dengan kapasitas rendah. Berdasarkan fakta tersebut, penelitian ini bertujuan untuk mengklasifikasikan citra gerakan Bahasa Isyarat Arab (*ArSL*) menggunakan arsitektur *Convolutional Neural Network (CNN)* dengan model *pre-trained EfficientNetB1*. Dengan menggunakan teknik *transfer learning*, model yang telah dilatih pada *dataset* besar dapat digunakan untuk menyesuaikan dengan *dataset* yang lebih kecil dan spesifik. Sehingga dapat mengurangi jumlah waktu yang dihabiskan untuk pelatihan dan mengurangi risiko *overfitting*. Dalam penelitian ini, proses *transfer learning* diterapkan dengan mengganti lapisan atas *EfficientNetB1* yang telah dilatih pada *dataset ImageNet* dengan lapisan yang sesuai dengan jumlah kelas dalam *dataset* bahasa isyarat. Selain itu, dilakukan *fine-tuning* dengan membekukan 30% dan 50% lapisan awal untuk mempertahankan fitur dasar sambil melatih ulang lapisan atas. Model *pre-trained EfficientNetB1* dipilih karena kemampuan untuk menyeimbangkan akurasi tinggi dan efisiensi komputasi melalui pendekatan *compound scaling*, yang secara efisien dapat meningkatkan skala model dalam hal resolusi gambar, kedalaman jaringan, dan lebar jaringan. *EfficientNetB1* dapat menghemat sumber daya komputasi sekaligus menjaga performa model tetap optimal.

Sistem yang dikembangkan dalam penelitian ini memiliki keunggulan dalam hal efisiensi dan akurasi. Meskipun model seperti *MobileNet* dan *VGGNet* lebih cepat untuk menginferensi pada perangkat dengan sumber daya terbatas, mereka tidak mencapai akurasi setinggi *EfficientNetB1*. Model seperti *Inception*, *ResNet*, dan *Xception* memiliki kinerja yang baik, tetapi dengan penggunaan sumber daya yang lebih besar. Sistem ini mengatasi kekurangan teknologi sebelumnya dengan memberikan solusi yang lebih seimbang antara kinerja dan efisiensi. Selain itu, sistem ini dapat beroperasi dengan baik pada perangkat dengan spesifikasi rendah. Dengan demikian, penelitian ini tidak hanya meningkatkan teknologi saat ini tetapi juga menawarkan inovasi yang relevan untuk pengembangan sistem pengenalan bahasa isyarat yang lebih baik.

## II. METODE PENELITIAN

Penelitian ini mengusulkan penggunaan metode *Convolutional Neural Network* dengan model arsitektur *EfficientNetB1* dan menerapkan teknik *transfer learning* untuk mengklasifikasikan citra gerakan bahasa isyarat Arab. Pada bagian ini menjelaskan tentang alur metode penelitian seperti yang ditunjukkan pada Gambar 1.



Gambar 1. Alur Metode Penelitian

Proses penelitian dimulai dengan pembuatan dataset huruf hijaiyah pada bahasa isyarat Arab. Kemudian dataset akan melalui tahap *pre-processing* dan akan dibagi menjadi tiga bagian: untuk tujuan pelatihan, validasi, dan pengujian. Dataset pelatihn dan validasi akan digunakan untuk memulai proses pelatihan dengan menggunakan arsitektur *EfficientNetB1*. Dataset pengujian akan digunakan untuk mengevaluasi hasil pelatihan dan mendapatkan metrik *precision*, *recall*, *accuracy*, dan *f1-score* dari model yang dikembangkan [14].

#### A. Dataset

Dalam penelitian ini, pengumpulan data primer dilakukan dengan menggunakan teknik observasi. Pengambilan citra bentuk tangan huruf hijaiyah pada bahasa isyarat Arab didokumentasikan menggunakan kamera *smartphone*, menghasilkan citra RGB (*Red Green Blue*) dengan rasio 3:4 dalam format *\*jpg*. Data ini dikumpulkan dari 20 partisipan, mencakup 28 huruf hijaiyah masing-masing terdiri dari 100 citra, sehingga total dataset mencapai 2800 citra. Penelitian ini tidak disertakan huruf ‘*alif lam*’ dan ‘*hamzah*’ karena huruf tersebut pada penerapannya perlu ada pergerakan dinamis untuk mengisyaratkan huruf tersebut. Gambar 2 menunjukkan citra huruf hijaiyah dalam Bahasa isyarat.



Gambar 2. Bahasa Isyarat Arab Huruf Hijaiyah

#### B. Pre-processing

*Pre-processing* adalah proses pengolahan citra digital yang dilakukan sebelum proses utama, dengan tujuan untuk meningkatkan dan memperbaiki kualitas citra. Proses ini membantu dalam ekstraksi fitur dan klasifikasi, sehingga dapat mengoptimalkan kinerja sistem dalam mengenali citra huruf hijaiyah pada bahasa isyarat Arab. Berikut ini merupakan proses yang dilakukan untuk klasifikasi citra huruf hijaiyah pada bahasa isyarat Arab :

##### 1. Resizing

Proses ini melibatkan mengubah ukuran citra menjadi 240 x 240 piksel, yang merupakan ukuran standar yang diperlukan oleh model *EfficientNetB1*. Tujuannya adalah untuk memastikan bahwa semua citra yang diproses memiliki ukuran yang sama, sehingga model dalam mengenali dan mempelajari pola dengan mudah. Dengan menstandarkan ukuran citra, model dapat bekerja dengan lebih efisien dan akurat karena data yang diterima memiliki dimensi sama, sehingga dapat mengurangi kemungkinan masalah.

##### 2. Normalization

Mengubah rentang nilai piksel dari 0-255 (skala warna *RGB* standar) menjadi rentang 0-1. Normalisasi ini penting untuk mempercepat proses pelatihan model dan meningkatkan konvergensi, karena model dapat belajar lebih cepat jika nilai input berada dalam rentang yang lebih kecil dan seragam.

##### 3. Dataset Splitting

Pada tahap ini, pembagian *dataset* dilakukan dengan metode ‘*stratify*’ dengan tujuan memastikan *label* didistribusikan secara seimbang di setiap *subset*. Hal ini memungkinkan model untuk belajar dari representasi data yang adil dan menghindari bias terhadap *label* yang jarang muncul.

Dataset dibagi menjadi tiga bagian utama, yaitu :

- Data *train*, bagian ini digunakan untuk melatih model, memungkinkan model belajar dari data yang sudah ada.
- Data *validation*, digunakan untuk mengatur parameter model dan mengevaluasi kinerja model secara berkala selama pelatihan. Hal ini membantu dalam proses memilih model yang terbaik dan mencegah *overfitting*.
- Data *test*, digunakan untuk mengevaluasi kinerja model setelah proses pelatihan selesai. Data ini tidak pernah diakses oleh model selama pelatihan, sehingga memberikan gambaran yang objektif tentang performa model.

Tabel 1 menunjukkan rincian proses pembagian data.

TABEL I  
 TABEL PEMBAGIAN DATA *TRAIN*, *VALIDATION*, *TEST*

No	Data	Presentase	Jumlah
1.	Data <i>Train</i>	70%	1960
2.	Data <i>Validation</i>	15%	420
3.	Data <i>Test</i>	15%	420

#### 4. Data Augmentation

Data *Augmentation* adalah teknik yang digunakan untuk memperluas dan meningkatkan variasi *dataset train* dengan menerapkan berbagai transformasi pada citra. Tujuan dari teknik ini adalah untuk membuat variasi citra baru dari citra yang ada. Hal ini dapat meningkatkan generalisasi model, mengurangi risiko *overfitting*, dan membuat model lebih kuat terhadap variasi yang belum ditemukan. Proses data *augmentation* ini dilakukan menggunakan library *Keras ImageDataGenerator*. Library ini memungkinkan parameter disesuaikan dengan kebutuhan citra huruf hijaiyah. Tabel 2 menunjukkan parameter yang digunakan dalam penelitian ini.

TABEL 2  
 PARAMETER DATA *AUGMENTATION*

No.	Augmentasi	Nilai
1.	<i>Horizontal_flip</i>	<i>true</i>
2.	<i>brightness_range</i>	0,75 , 1,3
3.	<i>Zoom_range</i>	0,2
3.	<i>Rotation_range</i>	40
4.	<i>Width_shift_range</i>	0,2
5.	<i>Height_shift_range</i>	0,2
6.	<i>Shear_range</i>	0,2
7.	<i>Fill_mode</i>	<i>nearest</i>

#### C. EfficientNetB1

Sistem model yang digunakan pada penelitian ini adalah *EfficientNetB1*. *EfficientNet* adalah salah satu model CNN yang telah di *pre-trained* oleh *Google* pada tahun 2019. Pengembangan *EfficientNet* difokuskan pada peningkatan kecepatan dan efisiensi komputasi. Model *EfficientNet* menunjukkan akurasi yang lebih tinggi dibandingkan dengan *pre-trained* model lainnya, meskipun memiliki jumlah parameter yang lebih sedikit [15]. Model *EfficientNet* merupakan salah satu teknik dalam *transfer learning* yang secara khusus dirancang untuk masalah pengenalan objek atau klasifikasi gambar. Dengan menggunakan *transfer learning*, risiko *overfitting* pada data pelatihan baru dapat diminimalkan, terutama jika data baru memiliki fitur yang serupa dengan tugas sebelumnya dan dataset asalnya lebih besar dari data baru tersebut [16]. *EfficientNet* menggunakan konvolusi *depthwise* dan *pointwise*. Penskalaan *EfficientNet* memiliki tiga dimensi, yaitu *depth*, *width*, dan *resolution*. *EfficientNet* terdiri dari tujuh blok dengan berbagai sub-blok. Koefisien gabungan  $\phi$  digunakan oleh *EfficientNet* untuk menskalakan lebar, kedalaman, dan resolusi jaringan [17]. Persamaan (1) hingga (3) berikut ini menunjukkan kedalaman, lebar, dan resolusi yang digunakan :

a. Kedalaman (*depth*)

$$d = \alpha\phi \tag{1}$$

b. Lebar (*width*)

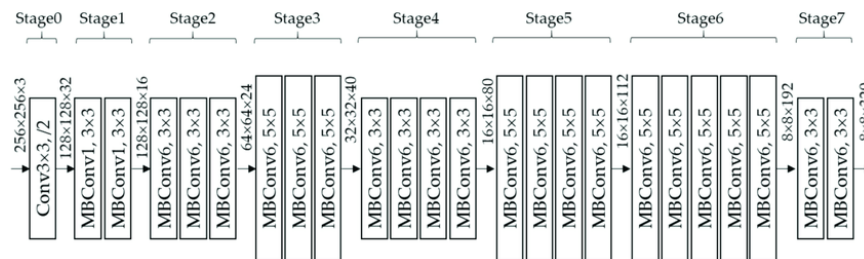
$$w = \beta\phi \tag{2}$$

c. Resolusi (*resolution*)

$$r = \gamma\phi \text{ s. t } \alpha, \beta, \gamma \approx 2 \tag{3}$$

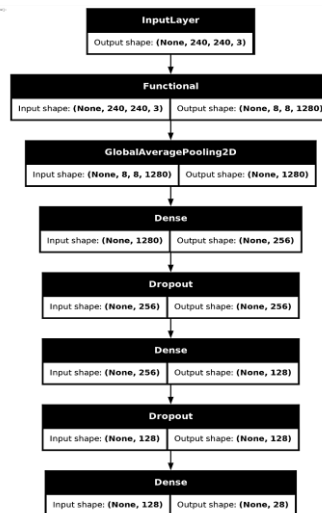
$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

Dimana konstanta  $\alpha, \beta, \gamma$  dapat ditentukan dengan melakukan pencarian *grid* terkecil. Koefisien  $\phi$  digunakan oleh pengguna untuk mengontrol jumlah sumber daya yang tersedia untuk penskalaan model, sedangkan  $\alpha, \beta, \text{ dan } \gamma$  untuk memutuskan bagaimana cara menentukan sumber daya tambahan untuk kedalaman, lebar, dan resolusi [18]. Penelitian ini menggunakan *EfficientNet-B1* sebagai *encoder* untuk ekstraksi fitur karena kemampuannya yang kuat dalam menangkap fitur spesifik bahasa isyarat Arab dengan efisiensi sumber daya. Dibandingkan model CNN lain seperti *MobileNet, VGGNet, Inception, ResNet, EfficientNet-B1* unggul dalam mengoptimalkan skala model melalui pendekatan *compound scaling*. Pendekatan ini memungkinkan model mencapai akurasi lebih akurat dengan menggunakan jumlah sumber daya komputasi yang lebih rendah. Oleh karena itu, *EfficientNet-B1* menjadi pilihan optimal untuk klasifikasi citra dalam pengenalan huruf hijaiyah pada bahasa isyarat Arab. Decoder yang digunakan sama dengan UNet asli. Struktur jaringan *EfficientNet-B1* ditunjukkan pada Gambar 3.



Gambar 3 Model *EfficientNetB1* [19]

Komponen dasar dari *EfficientNet-B1* adalah modul *MBConv*. Dalam modul *MBConv*, pertama-tama digunakan konvolusi  $1 \times 1$  untuk mengubah saluran fitur input, diikuti oleh konvolusi *depth-wise*, *pointwise*, normalisasi, dan fungsi aktivasi *ReLU*. Konvolusi *depthwise* digunakan untuk mengurangi jumlah komputasi dengan memproses setiap *channel* secara terpisah sebelum menggabungkan informasi melalui konvolusi *pointwise*. Selain itu, *EfficientNetB1* juga memanfaatkan blok *Squeeze-and-Excitation (SE)* untuk menyesuaikan bobot dari setiap fitur secara adaptif, meningkatkan representasi fitur yang lebih baik. Di bagian akhir arsitektur, lapisan *Global Average Pooling (GAP)* digunakan untuk mereduksi dimensi spasial dari fitur-fitur yang diekstraksi sebelum diproses melalui lapisan *fully connected* untuk melakukan klasifikasi akhir. Adapun usulan model *EfficientNetB1* pada penelitian ini ditunjukkan pada Gambar 4.



Gambar 4 Usulan Model *EfficientNetB1*

Proses pelatihan model ini menggunakan arsitektur jaringan *EfficientNet-B1* yang telah dilatih pada dataset *ImageNet*, tanpa lapisan *fully connected* di atasnya. Untuk mempertahankan representasi fitur yang telah dipelajari, lapisan-lapisan model dasar dibekukan. Untuk mengekstrak fitur, gambar dengan dimensi yang tepat dimasukkan sebagai input tensor. Data yang digunakan berukuran  $240 \times 240 \times 3$ , dengan 240 menunjukkan dimensi gambar yang telah dinormalisasi, dan 3 menunjukkan jumlah saluran warna pada gambar, yang berarti gambar berwarna dengan densitas 3 byte per piksel. Lapisan *Global Average Pooling* digunakan untuk mereduksi dimensi fitur spasial menjadi satu vektor global. Selanjutnya, lapisan *Dense* dan *Dropout* ditambahkan untuk meningkatkan kompleksitas representasi dan mencegah *overfitting*. Untuk melakukan evaluasi kinerja selama pelatihan, model dikompilasi dengan *optimizer Adam*, metrik akurasi, dan fungsi penurunan *crossentropy categorical*. Setelah definisi dan kompilasi selesai, model diperbarui melalui proses iteratif menggunakan data latih. Hal ini dilakukan untuk meningkatkan akurasi klasifikasi pada dataset yang digunakan dengan mengoptimalkan bobot-bobotnya berdasarkan *gradien* dari *loss function*. Berikut ini merupakan pengaturan *hyperparameter* pada model *Efficient-NetB1* ditunjukkan pada Tabel 3.

TABEL 3  
 HYPERPARAMETER MODEL PRE-TRAINED EFFICIENTNETB1

No	Hyperparameter	Nilai
1	Optimizer	Adam
2	Learning Rate	0,001
3	Loss function	Categorical Cross Entropy
4	Epoch	30
5	Dense_1	256
6	Dense_2	128
7	Dropout	0,5 dan 0,3
8	Batch Size	32

Tujuan dari penelitian ini adalah untuk mendapatkan hasil terbaik dari model *EfficientNet-B1* yang dikembangkan melalui pembuatan skenario pengujian. Skenario pengujian meliputi teknik *freeze* dan *unfreeze* pada *base model*. Selama proses *fine-tuning*, lapisan atas model dibuka untuk menyesuaikan bobot dengan dataset bahasa isyarat Arab, sementara lapisan dasar tetap dibekukan untuk mempertahankan pengetahuan dasar yang diperoleh dari pelatihan awal pada dataset *ImageNet*. Membekukan 30% dan 50% lapisan awal membantu model memanfaatkan fitur generik yang telah stabil dan mengurangi risiko *overfitting*, sedangkan membuka lapisan atas memungkinkan penyesuaian fitur dengan karakteristik dataset baru, sehingga meningkatkan akurasi klasifikasi. Pendekatan ini memastikan bahwa model dapat mempertahankan kualitas fitur yang ada sambil mengoptimalkan kinerja untuk tugas spesifik.

#### D. Evaluation

Tahap evaluasi merupakan proses penting dalam percobaan untuk menilai keakuratan hasil skenario pengujian. Evaluasi dilakukan dengan menganalisis hasil prediksi model dan membandingkannya dengan nilai sebenarnya dari data pengujian untuk menentukan sejauh mana model dapat mengklasifikasikan data dengan benar. Dalam penelitian ini, *confusion matrix* digunakan untuk mengevaluasi performa model dalam masalah klasifikasi dengan memberikan gambaran rinci tentang jumlah prediksi yang benar dan salah untuk setiap kelas. *Confusion matrix* membantu menghitung metrik seperti *precision*, *f1-score*, *accuracy* dan *recall* yang memberikan gambaran lebih mendalam tentang kekuatan dan kelemahan model [20]. *Precision* mengukur akurasi model dalam mengklasifikasikan data positif, sementara *F1-score* menggabungkan *precision* dan *recall* untuk memberikan gambaran lengkap tentang kinerja model secara keseluruhan. *Accuracy* menunjukkan persentase keseluruhan prediksi yang benar dari total sampel, dan *recall* mengindikasikan sejauh mana model dapat mengidentifikasi dan mengambil sampel positif dengan benar. Dengan menggunakan *confusion matrix*, penelitian ini bertujuan untuk mengukur dan menggambarkan kualitas model secara rinci dan obyektif, serta memahami kemampuan model untuk membedakan kelas-kelas yang berbeda dan mengidentifikasi area yang membutuhkan peningkatan atau perbaikan. Dengan demikian, evaluasi model menggunakan *confusion matrix* dan metrik *precision*, *f1-score*, *accuracy*, dan *recall* memungkinkan perbaikan iteratif dan peningkatan performa model.

Berikut ini adalah komponen dalam *confusion matrix* yang ditampilkan pada Tabel 4.

TABEL 4  
 CONFUSION MATRIX

Confusion Matrix		Predicted Class	
		Positif	Negatif
True Class	Positif	TP (True Positif)	FP (False Positif)
	Negatif	FN (False Negatif)	TN (True Negatif)

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \times 100\% \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (6)$$

$$F1 - Score = \frac{2 \times P \times R}{P + R} \quad (7)$$

Keterangan :

- TP (*True Positif*) = Data positif yang benar diprediksi sebagai positif oleh model.
- TN (*True Negatif*) = Data negatif yang benar diprediksi sebagai negatif oleh model.
- FP (*False Positif*) = Data negatif yang salah diprediksi sebagai positif oleh model.
- FN (*False Negatif*) = Data positif yang salah diprediksi sebagai negatif oleh model.

### III. HASIL DAN PEMBAHASAN

Pada bagian ini, hasil eksperimen dibagi menjadi 2 bagian utama yaitu hasil pelatihan dan validasi, serta hasil pengujian. Proses pelatihan dan validasi dilakukan menggunakan *platform Kaggle Notebook* dengan dukungan *GPU T4 X2*, yang memberikan kecepatan yang signifikan dalam melatih model. Pada tahapan ini, evaluasi dilakukan untuk memastikan bahwa model yang dibangun telah sesuai. Sebelum melakukan evaluasi, pengujian dilakukan pada model menggunakan 4 skenario yang berbeda. Dalam percobaan ini, teknik *unfreeze* dan *freeze* pada *base model* digunakan untuk mengubah parameter model *layers* yang dapat dilatih atau tidak. Kemudian dilakukan *Fine-tuning* yang bertujuan untuk meningkatkan akurasi model. Dalam proses *fine-tuning*, membekukan 30% dan 50% lapisan awal bertujuan untuk menemukan keseimbangan optimal antara mempertahankan pengetahuan umum dan menyesuaikan dengan fitur spesifik dari dataset baru. Untuk memulai proses *fine-tuning*, lapisan atas dari *base model* yang sebelumnya telah dibekukan dibuka. Dengan membuka lapisan ini, membuat model dapat menyesuaikan bobot pada lapisan-lapisan tersebut selama proses *fine-tuning*. Selanjutnya, beberapa lapisan dari *base model* dibekukan kembali untuk mengontrol proses *fine-tuning*. Dengan mencoba berbagai



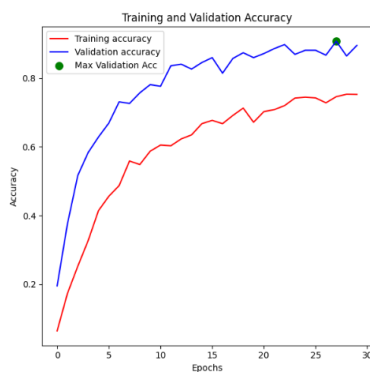
skenario ini, penelitian ini dapat menemukan konfigurasi yang memberikan performa terbaik untuk tugas klasifikasi bahasa isyarat Arab. Dengan demikian, hasil dari percobaan ini dapat digunakan untuk menentukan model mana yang memiliki performa yang lebih baik.

#### A. Hasil Pelatihan dan Validasi

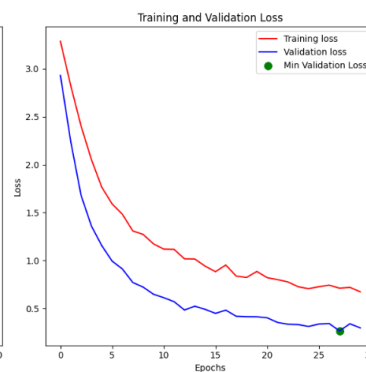
Pengimplementasian dilakukan menggunakan *Python* dengan berbagai *library* seperti *TensorFlow*, *Keras*, *Cv2*, *NumPy*, *Scikit-learn*, *Matplotlib*, *Seaborn* dan *Pandas*.

Pada skenario pertama seluruh lapisan dari base model dibekukan (*trainable false*). Tujuannya adalah untuk menggunakan semua pengetahuan umum yang telah dipelajari dari kumpulan *dataset* besar seperti *ImageNet* tanpa melakukan penyesuaian ulang pada *dataset* bahasa isyarat Arab.

Hasil pelatihan menunjukkan bahwa dengan menggunakan *transfer learning* dari model *pre-trained*, model mengalami peningkatan yang signifikan dalam akurasi. Dari akurasi yang sangat rendah pada awalnya menjadi lebih dari 74% pada data pelatihan setelah 30 *epoch*. Selain itu, akurasi pada data validasi juga meningkat secara signifikan dari sekitar 37.62% menjadi lebih dari 89.52%, dengan *loss* masing-masing 0.7080 dan 0.2970. Meskipun akurasi pelatihan lebih rendah dibandingkan dengan akurasi validasi, model tetap menunjukkan performa yang baik pada data validasi. Hal ini mengindikasikan bahwa meskipun lapisan-lapisan dasar tidak diubah dan hanya lapisan tambahan yang dilatih, model masih dapat memanfaatkan fitur-fitur umum yang dipelajari dari *dataset* besar seperti *ImageNet* untuk tugas spesifik klasifikasi bahasa isyarat Arab. Perbedaan antara akurasi pelatihan dan validasi dapat menunjukkan bahwa model berpotensi mengalami *underfitting* pada data pelatihan, tetapi performa validasi yang lebih baik menunjukkan kemampuan generalisasi yang baik pada data baru.

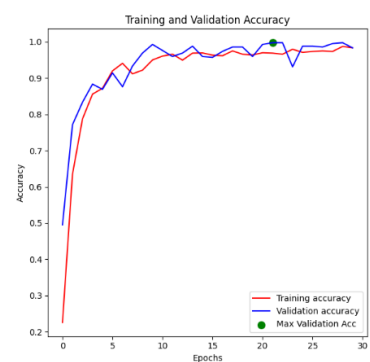


Gambar 5 Training and Validation Accuracy

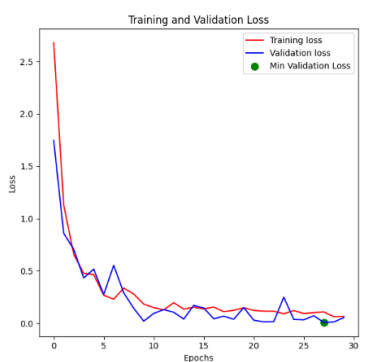


Gambar 6 Training and Validation loss

Dalam skenario kedua pelatihan ini, seluruh lapisan dari *base model* tidak dibekukan (*trainable true*), sehingga semua lapisan dapat dilatih ulang pada *dataset* bahasa isyarat Arab. Hal ini membuat seluruh lapisan model dapat sepenuhnya menyesuaikan dengan data baru.



Gambar 7 Training and Validation Accuracy



Gambar 8 Training and Validation loss

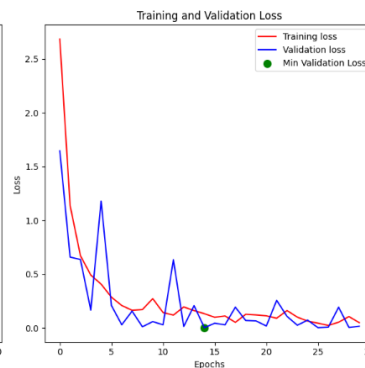
Hasil pelatihan menunjukkan akurasi tinggi yaitu 98.40% pada data pelatihan dan 98.33% pada data validasi, dengan *loss* yang rendah masing-masing 0.0611 dan 0.0537. Hasil ini menunjukkan bahwa model dapat menyesuaikan bobotnya dengan sangat baik pada *dataset* bahasa isyarat Arab. Hal ini mengindikasikan kemampuan model untuk generalisasi dengan baik tanpa *overfitting*, dan menunjukkan bahwa meng-*unfreeze* seluruh lapisan *base model* dapat menyesuaikan dengan optimal terhadap data spesifik. Pendekatan ini terbukti efektif untuk *dataset* yang cukup besar, di mana seluruh lapisan model dapat dilatih ulang untuk memaksimalkan kemampuan

adaptasi dan klasifikasi pada tugas tertentu.

Pada skenario ketiga, 30% lapisan awal dari *base model* dibekukan sementara dan sisanya tidak dibekukan. Dengan pendekatan ini, model dapat mempertahankan sebagian besar pengetahuan umum yang dipelajari dari *dataset* besar dengan menyesuaikan lapisan atas dari data spesifik bahasa isyarat Arab.



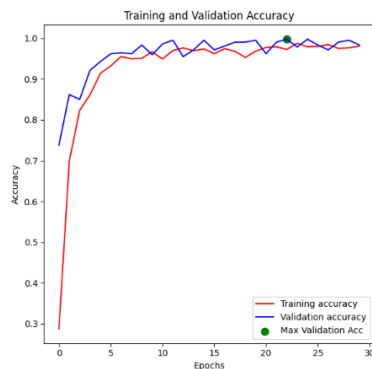
Gambar 9 Training and Validation Accuracy



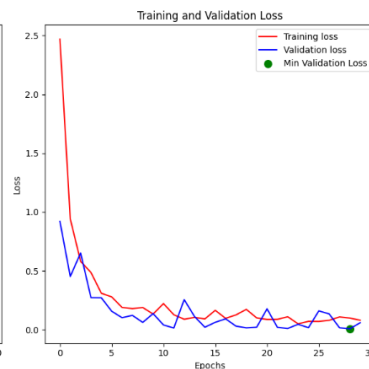
Gambar 10 Training and Validation loss

Hasil pelatihan menunjukkan bahwa model berkinerja dengan baik pada kedua data, dengan akurasi sebesar 98.95% pada data pelatihan dan 99.52% pada data validasi, dengan *loss* masing-masing 0.0481 dan 0.0163. Dengan membekukan 30% lapisan awal, model dapat mempertahankan pengetahuan umum yang telah dipelajari dari *dataset ImageNet*, sementara lapisan yang tidak dibekukan dapat menyesuaikan dengan fitur spesifik dari *dataset* bahasa isyarat Arab. Pendekatan ini membantu mengimbangi antara memanfaatkan fitur dasar yang sudah ada dan menyesuaikan model untuk tugas spesifik. Akurasi yang tinggi dan *loss* yang rendah pada data validasi menunjukkan bahwa model tidak hanya cocok dengan data pelatihan tetapi juga mampu generalisasi dengan baik pada data yang belum terlihat sebelumnya, menunjukkan efektivitas dari *fine-tuning* pada skenario ini.

Pada skenario keempat, lapisan awal dari *base model* dibekukan sebanyak 50%, dan lapisan sisanya dilatih ulang. Ini bertujuan untuk mempertahankan lebih banyak pengetahuan umum dari dataset besar, mengurangi risiko *overfitting*, namun masih memungkinkan penyesuaian yang cukup terhadap data spesifik dari bahasa isyarat Arab.



Gambar 11 Training and Validation Accuracy



Gambar 12 Training and Validation loss

Hasil pelatihan menunjukkan akurasi sebesar 98.01% pada data pelatihan dan 98.33% pada data validasi, dengan *loss* masing-masing 0.0825 dan 0.0600. Hasil ini menunjukkan bahwa meskipun 50% dari lapisan awal dibekukan, model tetap menunjukkan performa yang sangat baik pada data pelatihan dan validasi mengindikasikan bahwa model mampu menyesuaikan dengan baik terhadap *dataset* bahasa isyarat Arab tanpa mengalami *overfitting*.

Dari keempat skenario yang diuji, skenario ketiga dengan membekukan 30% lapisan awal, menunjukkan hasil yang paling optimal. Pendekatan ini menghasilkan akurasi tertinggi pada data pelatihan dan validasi serta *loss* terendah. Pendekatan ini berhasil mencapai keseimbangan yang ideal antara mempertahankan pengetahuan dasar dari *dataset ImageNet* dan menyesuaikan model dengan fitur spesifik dari data bahasa isyarat Arab. Hasil ini menunjukkan performa yang sangat baik secara keseluruhan, menjadikannya sebagai konfigurasi terbaik untuk tugas klasifikasi dalam konteks penelitian ini.

## B. Hasil Pengujian

Hasil pengujian model disajikan menggunakan *confusion matrix* dan metrik performa seperti *accuracy*, *precision*, *recall*, dan *f1-score*. Hasil dari *confusion matrix* akan digunakan untuk memahami kesalahan klasifikasi secara

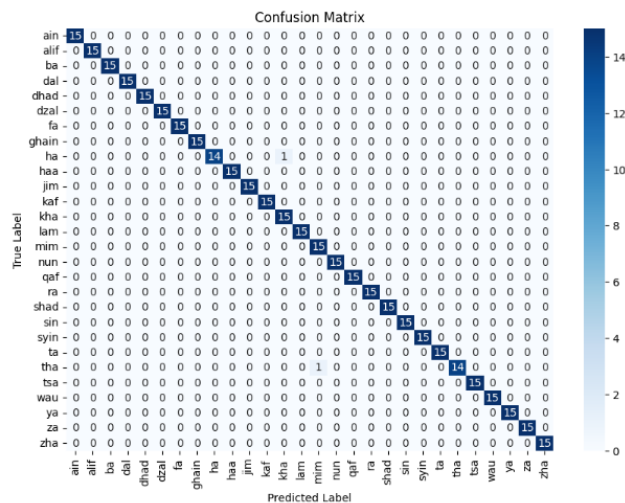
spesifik, sementara metrik kinerja akan memberikan informasi lengkap mengenai efektivitas model dalam memenuhi tujuan klasifikasi. Tabel 5 menunjukkan hasil implementasi dari model *EfficientNetB1* pada berbagai skenario pelatihan.

TABEL 5  
 HASIL IMPLEMENTASI

Model <i>EfficientNetB1</i>	Accuracy	Precision		Recall		F1-Score	
		macro	weighted	macro	weighted	macro	weighted
<i>Freeze</i>	88%	90%	90%	88%	88%	88%	88%
<i>Unfreeze</i>	98%	98%	98%	98%	98%	98%	98%
<i>Fine-tuning 30%</i>	100%	100%	100%	100%	100%	100%	100%
<i>Fine-tuning 50%</i>	99%	99%	99%	99%	99%	99%	99%

Dalam skenario pertama yaitu menerapkan teknik *Freeze*, di mana semua lapisan dalam *base model* dibekukan. Ini menunjukkan performa yang baik namun kurang optimal. Pada skenario kedua diterapkan teknik *Unfreeze*, di mana semua lapisan dapat dilatih ulang, model menunjukkan peningkatan signifikan dengan akurasi dan metrik lain sebesar 98%. Ini menandakan bahwa melatih seluruh lapisan *base model* memberikan hasil yang lebih baik. Dalam skenario ketiga dilakukan proses *Fine-Tuning* di mana 30% lapisan awal dibekukan, model mencapai performa terbaik dengan akurasi dan metrik lainnya 100%. Ini menunjukkan bahwa pendekatan ini berhasil memanfaatkan pengetahuan dasar dan dapat menyesuaikan dengan data spesifik. Terakhir, pada skenario keempat dilakukan proses *Fine-Tuning* di mana 50% lapisan awal dibekukan, model masih menunjukkan performa yang sangat baik dengan akurasi 99%, tetapi sedikit di bawah *fine-tuning 30%*. Secara keseluruhan, *fine-tuning 30%* lapisan awal memberikan hasil yang paling optimal, menunjukkan keseimbangan terbaik antara mempertahankan pengetahuan dasar dan menyesuaikan dengan data baru.

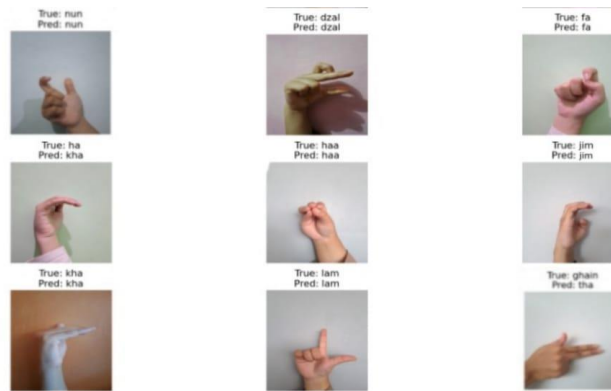
Proses klasifikasi dalam skenario ketiga digambarkan dalam *confusion matrix* pada Gambar 13. Secara keseluruhan, *confusion matrix* menunjukkan bahwa model klasifikasi yang digunakan memiliki kinerja yang sangat baik. Model mampu mengklasifikasikan sebagian besar sampel dengan akurat dan hanya membuat sedikit kesalahan.



Gambar 13 *Confusion matrix*

Dalam analisis ini, *confusion matrix* menunjukkan bahwa model secara keseluruhan memiliki performa yang sangat baik dalam mengklasifikasikan huruf isyarat Arab dengan sebagian besar kelas mencapai nilai *precision*, *recall*, dan *F1-score* yang mendekati 1.00, menandakan efektivitas dalam membedakan huruf isyarat Arab. Namun, terdapat beberapa kelas yang mengalami sedikit penurunan nilai metrik, yang mengindikasikan bahwa model mengalami kesulitan dalam membedakan antara beberapa huruf yang mirip atau memiliki fitur yang kurang jelas. Untuk memperbaiki kelemahan ini, hasil dari *confusion matrix* dapat digunakan untuk menambahkan lebih banyak data latih, meningkatkan kualitas gambar, serta menerapkan augmentasi data yang lebih beragam.

Selanjutnya, akan dilakukan prediksi menggunakan data *test* pada klasifikasi citra huruf hijaiyah menggunakan model yang telah dikembangkan. Pada Gambar 14 ditunjukkan hasil dari pengujian prediksi pada sebagian klasifikasi citra huruf hijaiyah. Terdapat beberapa kesalahan prediksi, seperti huruf "ha" yang diprediksi sebagai huruf "kha" dan huruf "ghain" yang diprediksi sebagai huruf "tha". Sementara itu, huruf "nun", "dzal", "fa", "haa", "jim", "kha", dan "lam" diprediksi dengan benar.



Gambar 14 Hasil Prediksi

Berdasarkan hasil pengujian prediksi, beberapa kesalahan klasifikasi signifikan ditemukan, terutama pada huruf "ha" yang diprediksi sebagai huruf "kha" dan huruf "ghain" yang diprediksi sebagai huruf "tha". Kesalahan ini dapat disebabkan oleh kesamaan visual antara huruf, di mana bentuk huruf yang mirip dapat membingungkan model, terutama dalam citra dengan kualitas rendah dan terbatasnya jumlah *dataset*. Selain itu, variasi dan kualitas citra pada data latih dapat mengurangi kemampuan model untuk membedakan huruf-huruf yang mirip. Kesalahan ini juga dapat disebabkan oleh teknik *preprocessing* dan augmentasi data yang kurang optimal, karena jika model tidak terlatih dengan baik maka akan mengalami kesulitan untuk mengenali variasi citra yang belum pernah dicoba pada model sebelumnya. Untuk mengatasi masalah ini, disarankan untuk meningkatkan kualitas dataset dengan mengumpulkan citra berkualitas tinggi yang mencakup berbagai variasi sudut pengambilan dan kondisi pencahayaan. Penerapan teknik *preprocessing* yang lebih optimal seperti penghapusan *noise*, serta augmentasi data yang lebih beragam, dapat membantu model belajar dari variasi yang lebih luas. Dengan menerapkan langkah-langkah perbaikan ini, diharapkan model dapat mengurangi kesalahan prediksi dalam klasifikasi huruf hijaiyah dan meningkatkan akurasi secara keseluruhan.

Dalam penelitian ini, hasil klasifikasi menggunakan *pre-trained EfficientNetB1* menunjukkan performa yang optimal, terdapat beberapa perbedaan signifikan dengan penelitian lain yang menggunakan metode arsitektur CNN yang sama ataupun model arsitektur yang berbeda. Tabel 6 berikut menunjukkan perbandingan hasil penelitian.

TABEL 6  
 PERBANDINGAN HASIL PENELITIAN

Penelitian	Arsitektur Model	Jenis Dataset	Akurasi	Keterangan
Bani et al. (2023) [4]	<i>MobileNetV2, VGG16, InceptionV3, ResNet50V2, ResNet152, Xception</i>	54,049 citra <i>ArSL</i>	100% ( <i>InceptionV3</i> )	Ukuran dataset yang besar dan lebih beragam dapat menghasilkan akurasi yang lebih tinggi.
Sundari et al. (2023) [8]	<i>EfficientNet-B0</i>	Citra kulit kering, normal, dan kombinasi.	100%	Model yang lebih sederhana dapat lebih mudah disesuaikan dengan data yang tersedia.
Wardana et al. (2021) [9]	<i>EfficientNet-B4</i>	87,000 citra <i>ASL</i>	99.81%	Ukuran dataset yang jauh lebih besar dan teknik pelatihan <i>Cyclical Learning Rate</i> yang diterapkan dapat meningkatkan akurasi.
Brianorman dan Munir (2023) [12]	<i>MnetV2, VGG16, Xception, ResNet50</i>	6.200 citra <i>ArSL</i>	99.85% ( <i>MnetV2, VGG16, Xception</i> )	Tidak melakukan <i>preprocessing</i> citra sebelum pelatihan. Selain itu, <i>dataset</i> citra bahasa isyarat masih bersumber dari 1 orang. Namun, implementasi hasil pembelajaran <i>MnetV2</i> menunjukkan kinerja yang baik untuk pengenalan bentuk jari secara <i>real-time</i> .

Meskipun *EfficientNetB1* juga menunjukkan kinerja yang optimal, perbedaan jenis citra, ukuran dataset, dan teknik pelatihan dapat menjelaskan perbedaan hasil akurasi dan prediksi. Dengan menerapkan arsitektur *EfficientNetB1* untuk klasifikasi bahasa isyarat Arab, penelitian ini menawarkan perspektif baru untuk pengembangan sistem pengenalan bahasa isyarat dengan menggunakan model yang lebih kompleks dan teknik

*fine-tuning* yang telah teruji dapat membantu memperluas pemahaman kita tentang kinerja *EfficientNetB1* dalam berbagai konteks.

#### IV. KESIMPULAN

Penelitian ini menunjukkan bahwa penggunaan *Convolutional Neural Network* (CNN) dengan model *pre-trained EfficientNetB1* dan teknik *Transfer Learning* terbukti menghasilkan performa yang optimal dalam tugas klasifikasi Bahasa Isyarat Arab (*ArSL*), khususnya dalam mengenali huruf Hijaiyah. Hasil percobaan yang telah dilakukan pada empat skenario menghasilkan performa yang beragam. Skenario pertama dengan melakukan *freeze* pada *base model* menghasilkan akurasi 74% pada data pelatihan dan data validasi 89.52%. Skenario kedua dengan menerapkan *unfreeze* pada *base model* mendapatkan akurasi tinggi yaitu 98.40% pada data pelatihan dan 98.33% pada data validasi. Skenario ketiga dengan membekukan 30% lapisan awal model menghasilkan akurasi 98.95% pada data pelatihan dan 99.52% pada data validasi. Skenario keempat dengan membekukan 50% lapisan awal model mendapatkan akurasi sebesar 98.01% pada data pelatihan dan 98.33% pada data validasi.

Hal tersebut menunjukkan bahwa skenario ketiga dengan membekukan 30% lapisan awal selama *fine-tuning* memberikan hasil yang paling optimal. Pada data pengujian model skenario ketiga mencapai performa terbaik dengan *accuracy*, *precision*, *recall*, dan *f1-score* sebesar 100%. Hasil *fine-tuning* dengan membekukan 30% lapisan awal lebih baik daripada 50% karena mencapai keseimbangan optimal antara mempertahankan pengetahuan umum dari dataset besar seperti *ImageNet* dan menyesuaikan dengan fitur spesifik data bahasa isyarat Arab. Membekukan 30% lapisan awal membuat banyak lapisan untuk belajar karakteristik unik dari data baru, meningkatkan akurasi dan kinerja model secara keseluruhan. Sebaliknya, membekukan 50% lapisan awal mengurangi fleksibilitas model, sehingga performanya sedikit lebih rendah. Hasil ini menunjukkan bahwa *fine-tuning* dapat meningkatkan kemampuan model untuk mengklasifikasikan gambar dengan lebih akurat dan konsisten. Dengan demikian, sistem pengenalan huruf Hijaiyah yang dikembangkan tidak hanya akurat dalam identifikasi tetapi juga adaptif terhadap variasi data.

Namun, tantangan seperti kesamaan visual antar huruf dan kualitas gambar yang buruk dapat menyebabkan kesalahan prediksi pada huruf tertentu. Oleh karena itu, untuk meningkatkan prediksi model, kualitas dataset dan metode pemrosesan gambar harus terus diperbaiki. Secara keseluruhan, penggunaan *EfficientNetB1* dengan teknik *transfer learning* dan penyesuaian *fine-tuning* menunjukkan potensi besar dalam membangun sistem pengenalan huruf Hijaiyah yang efektif dan akurat, sehingga dapat membantu orang dengan gangguan pendengaran untuk belajar huruf hijaiyah melalui bahasa isyarat Arab dengan mudah.

#### DAFTAR PUSTAKA

- [1] U. Hasanah and others, "Desain Pembelajaran Pendidikan Agama Islam bagi Anak Tunarungu di SLB Negeri 01 Jakarta," Jakarta: Fakultas Ilmu Tarbiyah dan Keguruan UIN Syarif Hidayatullah Jakarta, 2021.
- [2] M. Al-Barham, A. Jamal, and M. Al-Yaman, "Design of Arabic Sign Language Recognition Model," *arXiv preprint arXiv:2301.02693*, 2023.
- [3] S. Rahmat, S. A. Rahman, N. H. Tukiran, R. Musa, N. A. Othman, and A. A. A. Dzulkarnain, "Development of hearing impairment inventory for religious duties of Muslim adult," *Medical Journal of Malaysia*, vol. 76, no. 2, pp. 205–211, 2021.
- [4] Q. Bani Baker, N. Alqudah, T. Alsmadi, and R. Awawdeh, "Image-Based Arabic Sign Language Recognition System Using Transfer Deep Learning Models," *Applied Computational Intelligence and Soft Computing*, vol. 2023, no. 1, p. 5195007, 2023.
- [5] R. M. Duwairi and Z. A. Halloush, "Automatic recognition of Arabic alphabets sign language using deep learning," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 12, no. 3, 2022.
- [6] B. Y. AlKhuraym, M. M. Ben Ismail, and O. Bchir, "Arabic sign language recognition using lightweight cnn-based architecture," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, 2022.
- [7] S. Hayani, M. Benaddy, O. El Meslouhi, and M. Kardouchi, "Arab sign language recognition with convolutional neural networks," in *2019 International conference of computer science and renewable energies (ICCSRE)*, 2019, pp. 1–4.
- [8] N. A. Sundari, R. Magladena, and S. Saidah, "Klasifikasi Jenis Kulit Wajah Menggunakan Metode Convolutional Neural Network (CNN) Efficientnet-B0," *eProceedings of Engineering*, vol. 9, no. 6, 2023.
- [9] B. K. Wardana, E. Rachmawati, and T. A. B. Wirayuda, "Pengenalan Gestur Tangan Statis Menggunakan CNN dengan Arsitektur Efficient-Net B4," *eProceedings of Engineering*, vol. 8, no. 2, 2021.
- [10] W. R. PERDANI, R. MAGDALENA, and N. O. R. K. C. PRATIWI, "Deep Learning untuk Klasifikasi Glaukoma dengan menggunakan Arsitektur EfficientNet," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 10, no. 2, p. 322, 2022.
- [11] M. E. Al Rivan and S. Hartoyo, "Klasifikasi Isyarat Bahasa Indonesia Menggunakan Metode Convolutional Neural Network," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 2, pp. 364–373, 2022.
- [12] R. M. Y. Brianorman and R. Munir, "Perbandingan Pre-Trained CNN: Klasifikasi Pengenalan Bahasa Isyarat Huruf Hijaiyah," *Jurnal Sistem Informasi Bisnis*, vol. 13, no. 1, pp. 52–59, 2023.
- [13] L. Arisandi and B. Satya, "Sistem Klarifikasi Bahasa Isyarat Indonesia (Bisindo) Dengan Menggunakan Algoritma Convolutional Neural Network," *Jurnal Sistem Cerdas*, vol. 5, no. 3, pp. 135–146, 2022.
- [14] N. K. D. A. Putri, A. Luthfiarta, and P. L. W. E. Putra, "OPTIMIZING BUTTERFLY CLASSIFICATION THROUGH TRANSFER LEARNING: FINE-TUNING APPROACH WITH NASNETMOBILE AND MOBILENETV2," *Jurnal Teknik Informatika (Jutif)*, vol. 5, no. 3, pp. 685–692, 2024.
- [15] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019, pp. 6105–6114.
- [16] F. Zhuang et al., "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.
- [17] M. A. Kzrak, Z. Müftüoğlu, and T. Yldrm, "Limitations and challenges on the diagnosis of COVID-19 using radiology images and deep learning," in *Data science for COVID-19*, Elsevier, 2021, pp. 91–115.

- [18] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, "Classification of remote sensing images using EfficientNet-B3 CNN model with attention," *IEEE access*, vol. 9, pp. 14078–14094, 2021.
- [19] Y. Jie *et al.*, "Combined multi-layer feature fusion and edge detection method for distributed photovoltaic power station identification," *Energies (Basel)*, vol. 13, no. 24, p. 6742, 2020.
- [20] A. Kulkarni, D. Chong, and F. A. Batarseh, "Foundations of data imbalance and solutions for a data democracy," in *Data democracy*, Elsevier, 2020, pp. 83–106.