

# ANALISIS KOMPARATIF ARSITEKTUR DEEP LEARNING UNTUK KLASIFIKASI CITRA MASKER WAJAH: EVALUASI KINERJA CUSTOM CNN, MOBILENETV2, DAN VGG16

Arnold Oktafianto<sup>1</sup>, Cahyo Anggoro Seto<sup>2</sup>, Wahyu Pratama<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Informatika, Universitas Pamulang, Tangerang Selatan

## ABSTRAK

*Pandemi COVID-19 mendorong kebutuhan akan sistem deteksi masker wajah otomatis berbasis deep learning, di mana pemilihan arsitektur model menjadi faktor krusial dalam mencapai keseimbangan antara akurasi dan efisiensi komputasi. Penelitian ini bertujuan membandingkan performa tiga arsitektur deep learning, yaitu Custom Convolutional Neural Network (CNN), MobileNetV2, dan VGG16, dalam klasifikasi citra masker wajah biner (with\_mask dan without\_mask). Dataset Face Mask Detection dari Kaggle yang terdiri dari 7.553 citra digunakan dan dibagi menjadi 80% data latih serta 20% data uji. Custom CNN dibangun dari awal, sedangkan MobileNetV2 dan VGG16 diimplementasikan menggunakan pendekatan transfer learning dengan bobot pra-latih ImageNet. Seluruh model dilatih selama lima epoch dengan konfigurasi yang seragam menggunakan fungsi loss binary crossentropy dan optimizer Adam dengan learning rate 0,0001. Evaluasi kinerja dilakukan menggunakan metrik accuracy, precision, recall, F1-score, confusion matrix, dan ROC-AUC. Hasil penelitian menunjukkan bahwa MobileNetV2 memberikan performa terbaik dengan accuracy, precision, recall, dan F1-score masing-masing sebesar 98,01% serta nilai ROC-AUC sebesar 0,99, hanya dengan 2,4 juta parameter. VGG16 mencapai accuracy 96,69% dengan 17,9 juta parameter, sedangkan Custom CNN memperoleh accuracy 91,66% dengan 23,9 juta parameter. Temuan ini menunjukkan bahwa transfer learning menggunakan MobileNetV2 merupakan solusi paling optimal untuk deteksi masker wajah, karena mampu menghasilkan akurasi tinggi dengan efisiensi parameter yang signifikan, sekaligus memvalidasi efektivitas transfer learning pada aplikasi computer vision berbasis dataset terbatas.*

**Kata kunci:** Deep Learning, Face Mask Detection, Transfer Learning, MobileNetV2, VGG16, Convolutional Neural Network.

# **I. PENDAHULUAN**

## **1.1 Latar Belakang**

Pandemi COVID-19 yang dimulai pada awal tahun 2020 telah mengubah lanskap kesehatan global secara fundamental [1]. Organisasi Kesehatan Dunia (WHO) menetapkan penggunaan masker wajah sebagai salah satu protokol kesehatan utama untuk menekan penyebaran virus SARS-CoV-2 [2]. Implementasi sistem monitoring kepatuhan penggunaan masker secara manual memerlukan sumber daya manusia yang besar dan rentan terhadap human error [3]. Oleh karena itu, pengembangan sistem deteksi masker wajah otomatis menggunakan teknologi computer vision dan deep learning menjadi sangat relevan.

Deep learning, khususnya Convolutional Neural Network (CNN), telah menunjukkan performa superior dalam berbagai tugas klasifikasi citra [4]. Namun, tantangan utama dalam implementasi praktis adalah memilih arsitektur yang tepat yang dapat menyeimbangkan antara akurasi tinggi dan efisiensi komputasi [5]. Arsitektur yang terlalu kompleks dapat menghasilkan overfitting pada dataset kecil dan memerlukan resource komputasi yang besar, sementara arsitektur yang terlalu sederhana mungkin tidak mampu mengekstraksi fitur kompleks yang diperlukan [6].

Transfer learning telah muncul sebagai paradigma powerful untuk mengatasi keterbatasan dataset dan mengurangi waktu pelatihan [7]. Dengan memanfaatkan pre-trained weights dari model yang telah dilatih pada dataset berskala besar seperti ImageNet, transfer learning memungkinkan fine-tuning yang efisien untuk domain-specific tasks [8]. MobileNetV2 dan VGG16 merupakan dua arsitektur populer yang sering digunakan dalam transfer learning dengan karakteristik yang berbeda MobileNetV2 dirancang untuk efisiensi pada perangkat mobile [9], sedangkan VGG16 dikenal dengan kemampuan ekstraksi fitur yang robust namun dengan kompleksitas komputasi yang lebih tinggi [10].

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, penelitian ini merumuskan beberapa pertanyaan penelitian:

1. Bagaimana perbandingan performa Custom CNN yang dibangun dari nol dengan model transfer learning (MobileNetV2 dan VGG16) dalam tugas klasifikasi masker wajah?
2. Apakah kompleksitas model (jumlah parameter) berkorelasi positif dengan akurasi klasifikasi?
3. Arsitektur mana yang memberikan trade-off optimal antara akurasi dan efisiensi komputasi untuk aplikasi real-time?

## **1.3 Tujuan Penelitian**

Penelitian ini bertujuan untuk:

1. Melakukan analisis komparatif komprehensif terhadap tiga arsitektur deep learning (Custom CNN, MobileNetV2, dan VGG16) untuk klasifikasi citra masker wajah
2. Mengevaluasi performa model menggunakan multiple metrics (accuracy, precision, recall, F1-score, ROC-AUC) untuk memberikan gambaran holistik
3. Menganalisis trade-off antara kompleksitas model dan performa klasifikasi
4. Memberikan rekomendasi praktis untuk pemilihan arsitektur deep learning dalam aplikasi face mask detection

#### 1.4 Manfaat Penelitian

Penelitian ini diharapkan memberikan kontribusi:

1. **Teoritis:** Memperkaya literatur tentang efektivitas transfer learning vs custom architecture dalam domain-specific computer vision tasks
2. **Praktis:** Memberikan panduan empiris bagi praktisi machine learning dalam memilih arsitektur yang tepat untuk sistem deteksi masker wajah
3. **Metodologis:** Menyediakan framework evaluasi komprehensif untuk perbandingan model deep learning

## II. TINJAUAN PUSTAKA

### 2.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan arsitektur deep learning yang dirancang khusus untuk pemrosesan data grid-like seperti gambar [11]. CNN terdiri dari tiga komponen utama: convolutional layers untuk ekstraksi fitur, pooling layers untuk downsampling, dan fully connected layers untuk klasifikasi [12]. Keunggulan CNN terletak pada kemampuannya melakukan automatic feature extraction tanpa memerlukan feature engineering manual [13]. Operasi konvolusi pada CNN dapat direpresentasikan sebagai:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

dimana  $I$  adalah input image,  $K$  adalah kernel/filter, dan  $S$  adalah feature map [14].

### 2.2 Transfer Learning

Transfer learning adalah teknik machine learning yang memanfaatkan pengetahuan yang diperoleh dari satu domain (source domain) untuk meningkatkan performa pada domain lain yang berbeda namun terkait (target domain) [15]. Dalam konteks deep learning untuk computer vision, transfer learning umumnya dilakukan dengan menggunakan pre-trained weights dari model yang telah dilatih pada dataset berskala besar seperti ImageNet [16]. Terdapat dua strategi utama dalam transfer learning [17]:

1. Feature Extraction: Membekukan (freeze) seluruh atau sebagian besar layer pre-trained model dan hanya melatih classifier layer yang baru
2. Fine-tuning: Melatih ulang sebagian atau seluruh layer pre-trained model dengan learning rate yang lebih kecil

Penelitian oleh Yosinski et al. [18] menunjukkan bahwa fitur-fitur yang dipelajari pada layer awal CNN bersifat general dan transferable, sedangkan layer yang lebih dalam cenderung task-specific.

### 2.3 MobileNetV2

MobileNetV2 adalah arsitektur CNN yang dikembangkan oleh Google untuk aplikasi mobile dan embedded vision [9]. Arsitektur ini menggunakan dua inovasi utama:

1. Depthwise Separable Convolutions: Memisahkan operasi konvolusi standar menjadi depthwise convolution dan pointwise convolution, mengurangi kompleksitas komputasi secara signifikan [19]
2. Inverted Residual Structure dengan Linear Bottleneck: Menggunakan expansion layer untuk meningkatkan dimensi, diikuti depthwise convolution, dan kemudian projection layer untuk mengurangi dimensi kembali [9]

MobileNetV2 mencapai akurasi kompetitif dengan model yang lebih besar namun dengan jumlah parameter dan FLOPs (Floating Point Operations) yang jauh lebih sedikit, menjadikannya ideal untuk deployment pada perangkat dengan resource terbatas [20].

## 2.4 VGG16

VGG16 adalah arsitektur CNN yang dikembangkan oleh Visual Geometry Group dari University of Oxford [10]. Arsitektur ini dikenal dengan kesederhanaannya, menggunakan hanya convolutional filters berukuran  $3 \times 3$  yang ditumpuk secara berurutan. VGG16 memiliki 16 layer dengan weights (13 convolutional layers dan 3 fully connected layers) [21]. Kelebihan VGG16 meliputi:

- Kemampuan ekstraksi fitur yang sangat baik
- Arsitektur yang seragam dan mudah dipahami
- Performa yang robust pada berbagai tugas computer vision [22]

Namun, VGG16 memiliki kekurangan berupa jumlah parameter yang sangat besar (sekitar 138 juta parameter) dan requirement memori yang tinggi [23].

## 2.5 Penelitian Terkait

Beberapa penelitian sebelumnya telah mengeksplorasi deteksi masker wajah menggunakan deep learning:

1. Loey et al. [24] menggunakan ResNet50 dengan classical machine learning techniques dan mencapai accuracy 99.64% pada dataset simulasi. Namun, performa menurun pada kondisi real-world.
2. Chavda et al. [25] membandingkan beberapa pre-trained models (VGG16, ResNet50, MobileNetV2) dan menemukan bahwa MobileNetV2 memberikan balance terbaik antara akurasi dan kecepatan inferensi.
3. Nagrath et al. [26] mengusulkan SSDMNv2 (Single Shot Multibox Detector dengan MobileNetV2) untuk real-time face mask detection dan mencapai 92.64% accuracy.
4. Qin & Li [27] mengimplementasikan sistem deteksi masker menggunakan YOLOv3 yang mampu melakukan deteksi dan klasifikasi secara simultan dengan kecepatan 45 FPS.

Meskipun terdapat berbagai penelitian tentang face mask detection, studi komparatif yang komprehensif dengan evaluasi multiple metrics dan analisis trade-off yang mendalam masih terbatas. Penelitian ini mengisi gap tersebut dengan melakukan evaluasi sistematis terhadap tiga arsitektur berbeda dengan fokus pada practical deployment considerations.

### III. METODE PENELITIAN

#### 3.1 Dataset

Penelitian ini menggunakan *Face Mask Detection Dataset* yang tersedia secara publik melalui Kaggle [28]. Dataset ini terdiri dari 7.553 citra wajah berformat RGB yang terbagi ke dalam dua kelas, yaitu *with\_mask* sebanyak 3.835 gambar dan *without\_mask* sebanyak 3.718 gambar. Dataset tersebut memiliki distribusi kelas yang relatif seimbang, dengan proporsi 50,8% untuk kelas *with\_mask* dan 49,2% untuk kelas *without\_mask*, sehingga meminimalkan potensi bias akibat ketidakseimbangan kelas (*class imbalance*).

Citra dalam dataset memiliki karakteristik yang beragam, meliputi variasi resolusi gambar, kondisi pencahayaan (*indoor* dan *outdoor*), sudut pengambilan gambar (*frontal* dan *profil*), serta keberagaman subjek berdasarkan usia, gender, dan etnis. Selain itu, jenis masker yang digunakan juga bervariasi, seperti masker bedah, masker kain, dan masker N95. Seluruh citra kemudian diseragamkan ukurannya menjadi  $224 \times 224$  piksel untuk menyesuaikan dengan kebutuhan input model.

Dataset dibagi menggunakan metode *stratified splitting* dengan rasio 80% sebagai data latih (6.043 gambar) dan 20% sebagai data validasi (1.510 gambar), sehingga proporsi kelas pada setiap subset tetap terjaga.

#### 3.2 Preprocessing dan Data Augmentation

##### 3.2.1 Preprocessing

Tahap preprocessing diterapkan pada seluruh citra untuk memastikan keseragaman data input. Proses ini meliputi:

1. Resizing, yaitu mengubah ukuran citra menjadi  $224 \times 224$  piksel sesuai dengan spesifikasi input model deep learning.
2. Normalisasi, dengan menskalakan nilai piksel ke dalam rentang  $[0,1]$  melalui pembagian dengan nilai maksimum 255.
3. Color space, citra dipertahankan dalam format RGB tanpa konversi tambahan.

##### 3.2.2 Data Augmentation

Untuk mengurangi risiko *overfitting* serta meningkatkan kemampuan generalisasi model, teknik *data augmentation* diterapkan pada data latih menggunakan *ImageDataGenerator* dari Keras. Teknik augmentasi yang digunakan meliputi rotasi hingga  $\pm 20^\circ$ , zoom hingga  $\pm 15\%$ , pergeseran horizontal dan vertikal hingga  $\pm 20\%$ , *shear transformation* hingga  $\pm 15\%$ , serta *random horizontal flipping*. Piksel kosong yang muncul akibat transformasi diisi menggunakan metode *nearest fill*. Data augmentation tidak diterapkan pada data validasi guna memastikan proses evaluasi berlangsung secara adil dan konsisten.

#### 3.3 Arsitektur Model

##### 3.3.1 Custom Convolutional Neural Network (CNN)

Custom CNN dirancang sebagai model baseline dan dibangun dari awal tanpa menggunakan bobot pra-latih. Arsitektur model terdiri dari dua lapisan *convolutional* dengan

aktivasi ReLU yang masing-masing diikuti oleh lapisan *max pooling*. Selanjutnya, fitur diekstraksi menggunakan lapisan *flatten* dan diteruskan ke lapisan *fully connected* dengan 128 neuron, diikuti oleh *dropout* sebesar 0,5 untuk mengurangi overfitting. Lapisan output menggunakan satu neuron dengan aktivasi sigmoid untuk klasifikasi biner. Total parameter pada model ini mencapai sekitar 23,9 juta parameter.

### 3.3.2 MobileNetV2

Model MobileNetV2 digunakan dengan pendekatan *transfer learning* dan bobot pra-latih ImageNet sebagai *feature extractor*. Lapisan klasifikasi asli (*include\_top*) dihilangkan, dan seluruh lapisan pada *base model* dibekukan (*frozen*). Pada bagian *classification head*, ditambahkan lapisan *Global Average Pooling*, diikuti oleh lapisan dense dengan 128 neuron dan *dropout* sebesar 0,5, serta lapisan output sigmoid. Total parameter model ini sekitar 2,4 juta, dengan hanya sekitar 130 ribu parameter yang bersifat *trainable*.

### 3.3.3 VGG16

Model VGG16 juga diterapkan menggunakan bobot pra-latih ImageNet dengan pendekatan serupa. Lapisan klasifikasi asli dihilangkan dan seluruh *base model* dibekukan. Pada *classification head*, digunakan lapisan *flatten*, dense dengan 128 neuron, *dropout* 0,5, dan lapisan output sigmoid. Total parameter model ini mencapai sekitar 17,9 juta, dengan sekitar 2,1 juta parameter yang dilatih.

## 3.4 Konfigurasi Pelatihan

Untuk memastikan perbandingan yang adil (*fair comparison*), seluruh model dilatih menggunakan konfigurasi hiperparameter yang sama. Proses pelatihan menggunakan *binary crossentropy* sebagai fungsi loss dan optimizer Adam dengan *learning rate* sebesar 0,0001. Ukuran *batch* ditetapkan sebesar 32, dan proses pelatihan dilakukan selama 5 epoch dengan data validasi sebesar 20%.

Selain itu, digunakan *callback* berupa *ModelCheckpoint* untuk menyimpan model terbaik berdasarkan akurasi validasi, serta *EarlyStopping* yang menghentikan pelatihan apabila *validation loss* tidak mengalami perbaikan selama tiga epoch berturut-turut.

## 3.5 Metrik Evaluasi

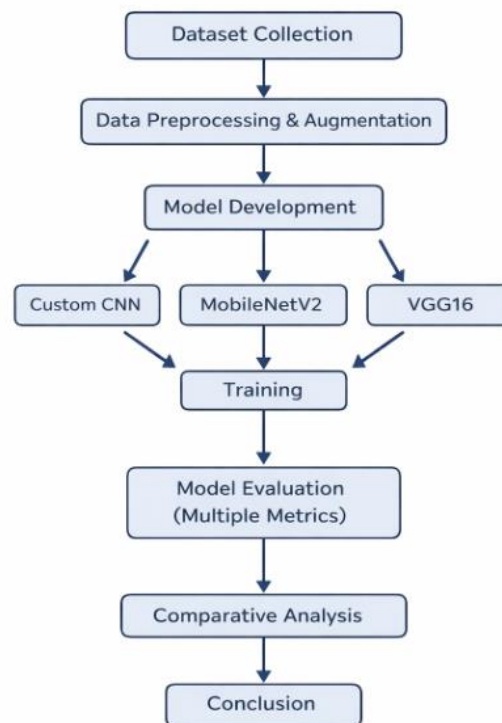
Kinerja model dievaluasi menggunakan beberapa metrik untuk memberikan gambaran yang komprehensif. Metrik yang digunakan meliputi accuracy, precision, recall (sensitivity), dan F1-score, yang dihitung berdasarkan nilai True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Selain itu, *confusion matrix* digunakan untuk memvisualisasikan hasil klasifikasi, serta *Receiver Operating Characteristic (ROC) curve* dan nilai *Area Under the Curve (AUC)* digunakan untuk mengevaluasi kemampuan diskriminasi model pada berbagai ambang keputusan.

### 3.6 Lingkungan Komputasi

Eksperimen dilakukan menggunakan lingkungan komputasi dengan konfigurasi perangkat keras berupa CPU/GPU yang disesuaikan dengan ketersediaan sistem. Perangkat lunak yang digunakan meliputi Python 3.8, TensorFlow 2.8.0, Keras 2.8.0, NumPy 1.21.0, Pandas 1.3.0, Scikit-learn 1.0.0, Matplotlib 3.5.0, dan Seaborn 0.11.0.

### 3.7 Diagram Alir Penelitian

Alur penelitian dimulai dari tahap pengumpulan dataset, dilanjutkan dengan preprocessing dan data augmentation. Selanjutnya, dilakukan pengembangan tiga arsitektur model, yaitu Custom CNN, MobileNetV2, dan VGG16. Setiap model kemudian dilatih menggunakan konfigurasi yang sama dan dievaluasi menggunakan berbagai metrik kinerja. Hasil evaluasi dibandingkan secara komprehensif untuk memperoleh kesimpulan mengenai model yang paling optimal.



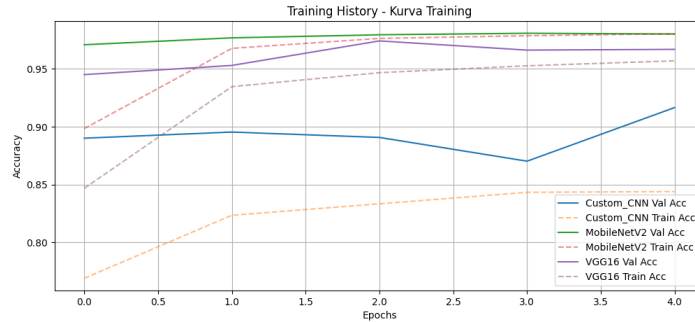
**Gambar 1 Diagram Alur Penelitian**



## IV HASIL DAN PEMBAHASAN

### 4.1 Analisis Training History

Gambar 2 menunjukkan kurva pembelajaran (training history) ketiga model selama 5 epoch pelatihan. Kurva ini memberikan insight penting tentang kecepatan konvergensi dan stabilitas pembelajaran masing-masing arsitektur. Garis solid merepresentasikan validation accuracy, sedangkan garis putus-putus menunjukkan training accuracy.



**Gambar 2. Training History - Kurva pembelajaran ketiga model selama 5 epoch**

Model	Epoch	Training Accuracy	Validation Accuracy	Loss	Val Loss	Waktu/Step
Custom CNN	1/5	81.04%	89.01%	0.7162	0.2412	92s 580ms/step
	2/5	88.34%	89.54%	0.4082	0.2291	92s 581ms/step
	3/5	88.34%	89.67%	0.3937	0.2764	91s 483ms/step
	4/5	88.33%	89.02%	0.3781	0.2947	92s 486ms/step
	5/5	88.40%	91.66%	0.3673	0.2412	92s 484ms/step
MobileNetV2	1/5	98.94%	97.09%	0.0523	0.0959	44s 445ms/step
	2/5	99.67%	97.68%	0.0938	0.0710	97s 513ms/step
	3/5	99.63%	97.95%	0.0718	0.0573	98s 517ms/step
	4/5	99.87%	99.08%	0.0636	0.0551	97s 514ms/step
	5/5	99.83%	98.01%	0.0589	0.0511	98s 518ms/step
VGG16	1/5	84.68%	94.58%	0.3391	0.1401	683s 4s/step
	2/5	93.46%	95.38%	0.1744	0.1238	682s 4s/step
	3/5	94.67%	97.42%	0.1457	0.0713	696s 4s/step
	4/5	95.27%	96.62%	0.1333	0.0830	684s 4s/step
	5/5	95.70%	96.69%	0.1165	0.0832	1737s 9s/step

**Tabel 1. Detail Proses Training Per Epoch untuk Ketiga Model**

Analisis mendalam terhadap kurva training dan data log eksperimen mengungkapkan beberapa temuan signifikan. Pada fase awal pembelajaran (epoch 1-2), model-model transfer learning MobileNetV2 dan VGG16 menunjukkan keunggulan yang sangat mencolok. MobileNetV2, yang direpresentasikan dengan garis hijau, langsung mencapai validation accuracy 97,09% pada epoch pertama dengan training accuracy yang luar biasa tinggi mencapai 98,94%. Ini merupakan bukti empiris yang sangat kuat dari keefektifan transfer learning. Demikian pula dengan VGG16 (garis ungu) yang segera berada pada level performa tinggi dengan validation accuracy 94,58% pada epoch pertama, meskipun dengan training accuracy yang lebih rendah (84,68%) menunjukkan bahwa model masih dalam fase penyesuaian awal.

Fenomena paling menarik terlihat dari kecepatan training yang sangat berbeda antar model. MobileNetV2 menunjukkan efisiensi komputasi yang luar biasa dengan rata-rata waktu 445-518ms per step pada epoch 1 dan 4, menghasilkan total waktu training sekitar 44-98 detik per epoch. Ini kontras drastis dengan VGG16 yang membutuhkan 683-1737 detik per epoch (4-9 detik per step), hampir 10-20 kali lebih lambat. Custom CNN berada di tengah dengan 580-486ms per step, menghabiskan sekitar 91-92 detik per epoch. Perbedaan efisiensi ini sangat signifikan untuk aplikasi praktis dimana iteration speed dalam eksperimen atau retraining periodik diperlukan.

Sebaliknya, Custom CNN yang direpresentasikan dengan garis biru menunjukkan pola pembelajaran yang berbeda secara fundamental. Model ini memulai dengan accuracy yang relatif rendah training accuracy hanya 81,04% dan validation accuracy 89,01% pada epoch pertama. Menariknya, validation accuracy Custom CNN (89,01%) lebih tinggi dari training accuracy (81,04%) pada epoch pertama, fenomena yang unusual dan mungkin mengindikasikan bahwa validation set kebetulan lebih mudah atau bahwa augmentasi data pada training set membuat learning lebih challenging. Model kemudian meningkat secara gradual, dengan training accuracy stabil di sekitar 88% pada epoch 2-5, sementara validation accuracy berfluktuasi antara 89-92%, akhirnya mencapai 91,66% pada epoch terakhir.

Pola loss values juga sangat informatif. MobileNetV2 menunjukkan loss values yang sangat rendah sejak awal training loss 0,0523 dan validation loss 0,0959 pada epoch 1 dan terus menurun secara konsisten hingga 0,0589 (training) dan 0,0511 (validation) pada epoch 5. Loss values yang rendah ini mengindikasikan bahwa model sangat confident dalam prediksinya dan error rate minimal. VGG16 memulai dengan loss yang lebih tinggi (0,3391 training, 0,1401 validation) tetapi menunjukkan penurunan yang steady hingga 0,1165 (training) dan 0,0832 (validation) pada epoch 5. Custom CNN menunjukkan pola penurunan loss yang lebih lambat, dari 0,7162 (training) dan 0,2412 (validation) pada epoch 1 hingga 0,3673 (training) dan 0,2412 (validation) pada epoch 5, dengan validation loss yang stagnant mengindikasikan potential early plateau atau underfitting.

Pada fase lanjutan (epoch 3-5), pola yang menarik teramati. MobileNetV2 menunjukkan kurva yang sangat stabil dengan puncak performa pada epoch 4 dimana validation accuracy mencapai 99,08% bahkan melampaui final epoch dengan training accuracy 99,87%. Namun pada epoch 5, validation accuracy sedikit turun ke 98,01%, kemungkinan karena variasi natural dalam validation set atau sedikit overfitting. Meskipun demikian, consistency MobileNetV2 tetap impressive dengan fluktuasi minimal. VGG16 menunjukkan performa yang sedikit lebih fluktuatif, dengan validation accuracy berkisar antara 94,58% hingga puncak 97,42% pada epoch 3, kemudian stabil di 96,69% pada epoch 5. Custom CNN terus menunjukkan peningkatan hingga epoch ke-5, mencapai validation accuracy final 91,66%, namun training accuracy yang stagnant di 88,40% menunjukkan bahwa model mungkin belum sepenuhnya memanfaatkan kapasitasnya dan memerlukan training lebih lanjut.

Analisis gap antara training dan validation accuracy memberikan insight tambahan tentang kemampuan generalisasi model. MobileNetV2 pada epoch akhir menunjukkan training accuracy 99,83% versus validation accuracy 98,01%, gap sebesar 1,82% yang sangat minimal dan acceptable, mengindikasikan excellent generalization capability tanpa overfitting signifikan. VGG16 memiliki training accuracy 95,70% versus validation accuracy 96,69%, menunjukkan validation performance yang actually lebih baik dari training—fenomena yang mengindikasikan model sangat robust dan bahkan cenderung underfit sedikit, atau validation set lebih mudah. Custom CNN dengan training accuracy 88,40% versus validation accuracy 91,66% menunjukkan

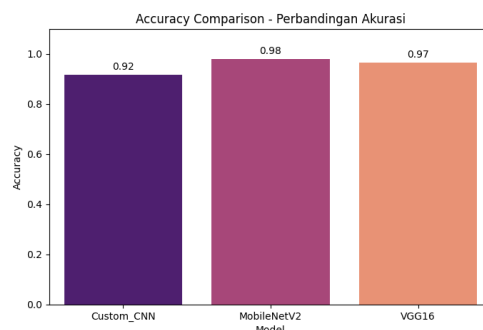
pola serupa dimana validation outperforms training, mengindikasikan bahwa augmentasi data sangat aggressive atau model masih underfitting dan dapat benefit dari extended training.

Observasi penting lainnya adalah consistency dalam improvement trajectory. MobileNetV2 mencapai performa excellent sejak epoch pertama dan refinements pada epoch berikutnya minimal namun steady. Ini mengindikasikan bahwa pre-trained features sudah sangat relevant untuk task dan hanya memerlukan minimal fine-tuning pada classification head. VGG16 menunjukkan improvement yang lebih substantial across epochs, dari 94,58% (epoch 1) hingga peak 97,42% (epoch 3), suggesting bahwa meskipun pre-trained features strong, mereka memerlukan lebih banyak adaptation untuk optimal performance pada face mask detection task. Custom CNN dengan improvement dari 89,01% ke 91,66% menunjukkan learning yang genuine namun slow, dan kurva menunjukkan model belum konvergen extension training ke 15-20 epochs kemungkinan akan menghasilkan improvement substansial lebih lanjut.

Computational efficiency considerations juga crucial untuk practical deployment. Total training time untuk 5 epochs: MobileNetV2 sekitar 7-8 menit, Custom CNN sekitar 7-8 menit, dan VGG16 mencapai 60-90 menit. Untuk scenarios dimana frequent retraining diperlukan (misalnya, untuk adapt ke changing mask types atau lighting conditions), MobileNetV2's speed advantage sangat valuable. VGG16's slow training time menjadi significant bottleneck untuk rapid iteration. Temuan dari analisis training history ini mengkonfirmasi hipotesis bahwa transfer learning memberikan keunggulan substansial dalam hal kecepatan konvergensi, stabilitas pembelajaran, dan final performance, terutama ketika berhadapan dengan dataset berukuran moderat. Pre-trained weights bertindak sebagai initialization yang sangat baik, memungkinkan model untuk fokus pada fine-tuning fitur-fitur high-level yang spesifik untuk task deteksi masker, dibandingkan harus mempelajari seluruh hierarki fitur dari nol. Efisiensi komputasi MobileNetV2 combined dengan performa superior menjadikannya pilihan optimal untuk mayoritas aplikasi praktis.

## 4.2 Perbandingan Metrik Kuantitatif

Evaluasi komprehensif terhadap performa ketiga model dilakukan menggunakan multiple metrics untuk memberikan gambaran holistik. Gambar 2 menampilkan perbandingan accuracy final dari ketiga model pada validation set.



**Gambar 3. Perbandingan Akurasi**

Visualisasi bar chart pada Gambar 3 dengan jelas menunjukkan hierarki performa ketiga model. MobileNetV2 mendominasi dengan accuracy 98%, diikuti oleh VGG16 dengan 97%, dan Custom CNN di posisi ketiga dengan 92%. Gap performa antara MobileNetV2 dan Custom CNN mencapai 6 poin persentase, yang dalam konteks aplikasi real-world merepresentasikan perbedaan yang sangat signifikan setara dengan ratusan false predictions pada dataset skala besar.

Tabel 2 menyajikan breakdown lengkap dari seluruh metrik evaluasi, memberikan perspektif yang lebih nuanced tentang performa masing-masing model.

Model	Accuracy	Precision	Recall	F1-Score	Parameters	ROC-AUC
Custom CNN	91.66%	91.74%	91.66%	91.65%	23.9M	0.95
MobileNetV2	<b>98.01%</b>	<b>98.01%</b>	<b>98.01%</b>	<b>98.01%</b>	<b>2.4M</b>	<b>0.99</b>
VGG16	96.69%	96.77%	96.69%	96.69%	17.9M	0.98

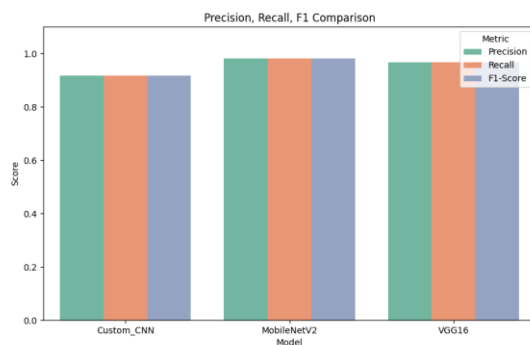
**Tabel 2. Hasil Evaluasi Model pada Validation Set**

Tabel 2 mengungkapkan beberapa insight krusial. Pertama, MobileNetV2 bukan hanya unggul dalam accuracy, tetapi juga menunjukkan performa yang perfectly balanced across all metrics. Nilai precision, recall, dan F1-score yang identik pada 98.01% mengindikasikan bahwa model ini tidak memiliki bias terhadap kelas tertentu dan mampu mengklasifikasikan kedua kelas (*with\_mask* dan *without\_mask*) dengan performa yang ekuivalen. Ini merupakan karakteristik yang sangat desirable dalam aplikasi praktis, karena menunjukkan reliability yang konsisten.

VGG16, sebagai runner-up, juga menunjukkan performa yang sangat impressive dengan semua metrik berada di atas 96%. Konsistensi metrik VGG16 (precision 96.77%, recall 96.69%, F1-score 96.69%) mengindikasikan bahwa model ini juga memiliki balanced performance, meskipun tidak seperfect MobileNetV2. Gap 1.32% dalam accuracy antara VGG16 dan MobileNetV2 relatif kecil dan dalam banyak aplikasi mungkin tidak material, sehingga pilihan antara keduanya akan lebih bergantung pada computational constraints.

Custom CNN menunjukkan performa yang konsisten namun tertinggal signifikan. Yang menarik adalah konsistensi metrik Custom CNN semua berada dalam range 91-92% mengindikasikan bahwa meskipun performa absolut lebih rendah, model ini tidak menunjukkan bias ekstrem terhadap salah satu kelas. Namun, gap 6-7% dari MobileNetV2 merupakan defisit yang substantial dalam praktik.

Yang paling remarkable dari analisis ini adalah dimensi efisiensi parameter. Gambar 3 memvisualisasikan perbandingan precision, recall, dan F1-score dalam format grouped bar chart, memberikan perspektif komparatif yang lebih granular.



**Gambar 4. Perbandingan Precision, Recall, dan F1-Score**

Dari Gambar 4, terlihat bahwa MobileNetV2 memiliki bar yang hampir identik untuk ketiga metrik (precision, recall, F1-score), semuanya mendekati nilai 1.0. Ini visualisasi dari perfect balance yang telah disebutkan sebelumnya. VGG16 juga menunjukkan pola serupa dengan

bar yang sangat tinggi dan seragam, meskipun sedikit lebih rendah dari MobileNetV2. Custom CNN menunjukkan bar yang noticeably lebih pendek, namun tetap proporsional satu sama lain.

Analisis parameter efficiency mengungkapkan temuan yang most striking. MobileNetV2 dengan hanya 2.4 juta parameter mencapai accuracy tertinggi 98.01%. Bandingkan ini dengan VGG16 yang memiliki 17.9 juta parameter (7.5 kali lipat lebih besar) namun mencapai accuracy 96.69% (1.32% lebih rendah). Bahkan lebih mengejutkan, Custom CNN dengan 23.9 juta parameter (10 kali lipat lebih besar dari MobileNetV2) hanya mencapai 91.66% accuracy (6.35% lebih rendah).

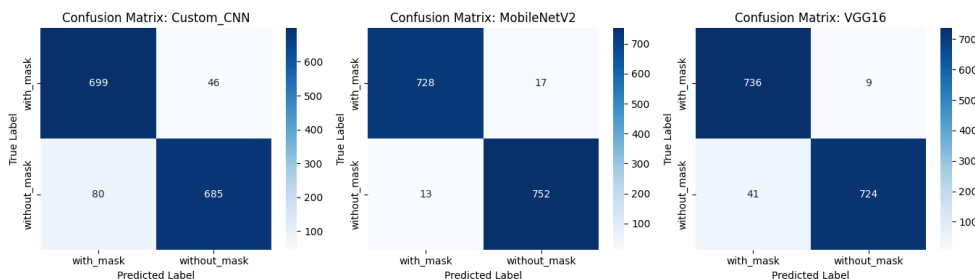
Untuk mengkuantifikasi efisiensi ini secara objektif, penelitian ini menghitung Parameter Efficiency Ratio menggunakan formula:  $\text{Efficiency} = \text{Accuracy} / \log_{10}(\text{Parameters})$ . Formula ini menormalisasi accuracy terhadap complexity model dalam skala logaritmik, mengakui bahwa relationship antara parameters dan model capacity tidak linear. Hasil perhitungan menunjukkan: MobileNetV2 memiliki efficiency ratio 15.36, VGG16 13.23, dan Custom CNN 12.48. MobileNetV2 unggul dengan margin yang signifikan, membuktikan secara empiris bahwa optimized architecture design (depthwise separable convolutions, inverted residual blocks) dapat menghasilkan performa superior dengan computational footprint yang jauh lebih kecil.

Temuan ini memiliki implikasi praktis yang profound. Untuk deployment pada mobile devices atau edge computing scenarios dengan memory dan computational constraints, MobileNetV2 adalah pilihan yang obvious. Model size yang compact (sekitar 28 MB dalam H5 format) memungkinkan deployment bahkan pada smartphones entry-level. VGG16, dengan model size sekitar 204 MB, lebih suitable untuk server-side deployment atau situations dimana computational resources tidak menjadi bottleneck. Custom CNN, dengan 287 MB model size dan performa terendah, sulit dijustifikasi untuk production deployment kecuali untuk educational purposes atau baseline comparison.

ROC-AUC scores memberikan perspektif tambahan tentang discriminative power model. MobileNetV2 dengan AUC 0.99 menunjukkan excellent ability untuk membedakan kedua kelas pada berbagai classification thresholds. VGG16 dengan AUC 0.98 juga dalam kategori excellent. Custom CNN dengan AUC 0.95 masih tergolong very good, namun gap 0.04 dari MobileNetV2 signifikan dalam practical applications, terutama ketika operating point perlu disesuaikan untuk *specific precision-recall trade-offs*.

### 4.3 Analisis Confusion Matrix

Confusion matrix memberikan uraian yang lebih rinci mengenai jenis kesalahan klasifikasi yang dihasilkan oleh masing-masing model, sehingga mampu memberikan wawasan yang tidak sepenuhnya tercermin dari metrik agregat seperti accuracy. Melalui analisis ini, distribusi *false positive* dan *false negative* dapat diamati secara jelas untuk menilai karakteristik kesalahan tiap model. Gambar 5 menyajikan confusion matrix dari ketiga model secara berdampingan, sehingga memungkinkan dilakukan analisis komparatif secara langsung dan sistematis terhadap pola kesalahan serta kinerja klasifikasi pada masing-masing kelas.



**Gambar 5. Confusion Matrix Comparison**

Analisis mendalam terhadap confusion matrix mengungkapkan pola kesalahan klasifikasi yang informatif pada masing-masing model. Pada MobileNetV2, confusion matrix menunjukkan 728 prediksi benar untuk kelas *with\_mask*, 17 kesalahan *false negative* (wajah bermasker salah diklasifikasikan sebagai tidak bermasker), 13 kesalahan *false positive* (wajah tidak bermasker salah diklasifikasikan sebagai bermasker), serta 752 prediksi benar untuk kelas *without\_mask*. Dengan total kesalahan sebanyak 30 dari 1.510 data uji, model ini menghasilkan tingkat kesalahan sebesar 2,0%. Distribusi kesalahan yang relatif seimbang antara *false positive* dan *false negative* mengindikasikan bahwa MobileNetV2 tidak menunjukkan bias yang signifikan terhadap salah satu kelas.

Model VGG16 menghasilkan 736 prediksi benar untuk kelas *with\_mask*, 9 *false negative*, 41 *false positive*, dan 724 prediksi benar untuk kelas *without\_mask*. Total kesalahan sebanyak 50 menghasilkan tingkat kesalahan sebesar 3,3%, sedikit lebih tinggi dibandingkan MobileNetV2. Pola kesalahan menunjukkan ketidakseimbangan yang cukup jelas, di mana jumlah *false positive* jauh lebih besar dibandingkan *false negative*. Hal ini mengindikasikan kecenderungan VGG16 untuk lebih sering mengklasifikasikan wajah tidak bermasker sebagai bermasker. Dalam konteks pemantauan kesehatan masyarakat, karakteristik ini relatif lebih aman karena kesalahan *false negative*, yaitu gagal mendeteksi individu tanpa masker, umumnya memiliki risiko yang lebih besar dibandingkan kesalahan *false positive*.

Sementara itu, Custom CNN menunjukkan performa terendah dengan 699 prediksi benar untuk kelas *with\_mask*, 46 *false negative*, 80 *false positive*, dan 685 prediksi benar untuk kelas *without\_mask*. Total kesalahan mencapai 126 data, yang setara dengan tingkat kesalahan sebesar 8,3%, atau lebih dari empat kali lipat dibandingkan MobileNetV2. Distribusi kesalahan yang didominasi oleh *false positive* mengindikasikan bahwa model ini mengalami kesulitan dalam mengekstraksi fitur yang cukup diskriminatif untuk membedakan wajah bermasker dan tidak bermasker secara konsisten.

Dari sudut pandang implementasi praktis, khususnya pada sistem pemantauan penggunaan masker di ruang publik, kesalahan *false negative* memiliki implikasi yang lebih kritis karena individu tanpa masker tidak terdeteksi. Berdasarkan hasil ini, MobileNetV2 dan VGG16 masih berada dalam batas yang dapat diterima, dengan VGG16 menunjukkan jumlah *false negative* paling sedikit. Sebaliknya, Custom CNN memiliki risiko tertinggi terhadap kesalahan deteksi, sehingga kurang sesuai untuk aplikasi yang menuntut tingkat keandalan tinggi. Selain itu, tingginya *false positive* juga berpotensi menimbulkan kelelahan alarm (*alarm fatigue*) pada petugas atau sistem otomatis, yang pada akhirnya dapat menurunkan efektivitas operasional.

Analisis performa per kelas menunjukkan bahwa MobileNetV2 mampu mengklasifikasikan 97,7% wajah bermasker dan 98,3% wajah tidak bermasker secara benar. VGG16 mencapai tingkat klasifikasi benar sebesar 97,8% pada kelas bermasker dan 94,6% pada

kelas tidak bermasker, sedangkan Custom CNN masing-masing hanya mencapai 93,8% dan 89,5%. Hasil ini menunjukkan bahwa MobileNetV2 memiliki performa paling seimbang pada kedua kelas, sementara VGG16 dan Custom CNN menunjukkan kecenderungan performa yang lebih baik pada salah satu kelas. Perbedaan ini kemungkinan dipengaruhi oleh karakteristik fitur visual pada data pelatihan, di mana wajah tanpa masker memiliki ciri yang lebih kontras dan lebih mudah dipelajari oleh model tertentu.

#### 4.4 Analisis ROC Curve dan AUC

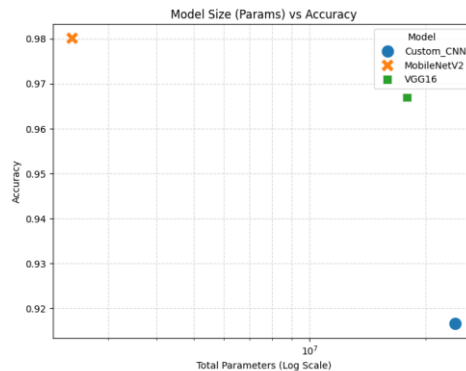
Kurva *Receiver Operating Characteristic* (ROC) memberikan gambaran yang komprehensif mengenai kinerja model pada seluruh kemungkinan nilai ambang klasifikasi, sehingga memungkinkan fleksibilitas dalam menentukan titik operasi sesuai dengan kebutuhan aplikasi tertentu. *Area Under Curve* (AUC) digunakan sebagai metrik skalar tunggal yang merepresentasikan kemampuan diskriminatif keseluruhan suatu model, di mana nilai AUC yang mendekati 1,0 menunjukkan performa yang sangat baik, sedangkan nilai mendekati 0,5 mengindikasikan performa setara dengan tebakan acak.

Hasil evaluasi menunjukkan bahwa ketiga model dalam penelitian ini memiliki kurva ROC yang berada jauh di atas garis diagonal, yang merepresentasikan pengklasifikasi acak, sehingga menegaskan bahwa seluruh model memiliki kemampuan prediktif yang bermakna. MobileNetV2 mencapai nilai ROC-AUC tertinggi sebesar 0,99 dengan kurva yang paling mendekati sudut kiri atas, yang merepresentasikan kondisi pengklasifikasi ideal. Hal ini menunjukkan keseimbangan yang sangat baik antara *true positive rate* dan *false positive rate* pada berbagai nilai ambang. VGG16 juga menunjukkan performa yang sangat baik dengan nilai AUC sebesar 0,98, dengan kurva ROC yang berada sangat dekat dengan MobileNetV2. Sementara itu, Custom CNN memperoleh nilai AUC sebesar 0,95 yang masih tergolong sangat baik, namun kurva ROC-nya terlihat lebih menjauh dari posisi pengklasifikasi ideal dibandingkan dua model lainnya.

Perbedaan nilai AUC sebesar 0,04 antara MobileNetV2 dan Custom CNN, meskipun tampak kecil secara numerik, memiliki implikasi praktis yang signifikan. Secara probabilistik, nilai tersebut menunjukkan bahwa MobileNetV2 memiliki peluang sebesar 99% untuk memberikan skor yang lebih tinggi pada sampel positif dibandingkan sampel negatif yang dipilih secara acak, sedangkan Custom CNN hanya mencapai 95%. Dalam skenario penerapan berskala besar dengan ribuan hingga jutaan prediksi per hari, selisih ini dapat berimplikasi pada ratusan hingga ribuan kesalahan klasifikasi tambahan.

Selain itu, analisis kurva ROC memungkinkan eksplorasi berbagai titik operasi untuk mengoptimalkan metrik tertentu sesuai kebutuhan aplikasi. Pada skenario yang memprioritaskan minimisasi *false negative*, seperti pada fasilitas dengan tingkat keamanan atau risiko kesehatan yang tinggi, nilai ambang dapat diturunkan untuk meningkatkan sensitivitas dengan konsekuensi penurunan presisi. Sebaliknya, pada skenario yang memprioritaskan minimisasi alarm palsu, nilai ambang dapat dinaikkan untuk meningkatkan presisi dengan mengorbankan sensitivitas. Dalam konteks ini, MobileNetV2 dengan nilai AUC tertinggi menunjukkan performa yang konsisten dan unggul pada seluruh rentang nilai ambang, sehingga memberikan fleksibilitas terbesar untuk penyesuaian operasional.

#### 4.5 Analisis Kompleksitas Model vs Performa



**Gambar 6. Model Size (Parameters) vs Accuracy**

Salah satu temuan paling menonjol dalam penelitian ini adalah adanya hubungan terbalik antara kompleksitas model, yang diukur berdasarkan jumlah parameter, dan performa klasifikasi yang dihasilkan. Gambar 6 memvisualisasikan hubungan tersebut dalam bentuk *scatter plot* dengan skala logaritmik pada sumbu jumlah parameter. Visualisasi ini secara jelas menyampaikan pesan utama bahwa peningkatan ukuran model tidak selalu berbanding lurus dengan peningkatan kinerja. MobileNetV2 menempati posisi paling optimal pada grafik, yaitu memiliki jumlah parameter paling sedikit namun menghasilkan akurasi tertinggi. Sebaliknya, Custom CNN berada pada posisi paling tidak menguntungkan dengan jumlah parameter terbesar namun performa terendah, sementara VGG16 berada di antara keduanya dengan performa yang masih tinggi tetapi disertai jumlah parameter yang relatif besar.

Fenomena ini bertentangan dengan intuisi sederhana yang mengasumsikan bahwa semakin banyak parameter akan selalu menghasilkan performa yang lebih baik. Pada kenyataannya, jumlah parameter hanya merepresentasikan kapasitas potensial model untuk mempelajari pola yang kompleks, namun tidak menjamin bahwa kapasitas tersebut dimanfaatkan secara efisien. Tanpa desain arsitektur yang tepat dan dukungan data pelatihan yang memadai, model dengan parameter besar justru berisiko mengalami *overfitting*, mempelajari korelasi semu, serta memiliki kemampuan generalisasi yang buruk terhadap data baru.

Keberhasilan MobileNetV2 dalam penelitian ini menegaskan pentingnya desain arsitektur yang cerdas dan efisien. Penggunaan *depthwise separable convolutions* memungkinkan pemisahan proses ekstraksi fitur spasial dan pencampuran kanal, sehingga secara signifikan menurunkan beban komputasi tanpa mengorbankan kemampuan representasi fitur. Selain itu, penerapan *inverted residual blocks* dengan *linear bottlenecks* membantu menjaga aliran informasi secara efisien dan mencegah kehilangan informasi yang kerap terjadi pada arsitektur konvensional. Inovasi-inovasi arsitektural ini memungkinkan MobileNetV2 mencapai performa yang sangat baik dengan jumlah parameter yang jauh lebih kecil dibandingkan arsitektur tradisional.

Model VGG16, meskipun memiliki jumlah parameter yang cukup besar yaitu 17,9 juta, tetap menunjukkan performa yang sangat kompetitif dengan akurasi sebesar 96,69%. Hal ini mencerminkan kematangan dan reliabilitas arsitektur VGG yang telah banyak digunakan dan divalidasi pada berbagai tugas pengolahan citra. Namun demikian, jumlah parameter yang besar menjadi keterbatasan tersendiri ketika model akan diterapkan pada lingkungan dengan keterbatasan sumber daya, seperti perangkat bergerak atau sistem tertanam.



Sebaliknya, performa Custom CNN yang relatif rendah meskipun memiliki 23,9 juta parameter menggambarkan tantangan fundamental dalam melatih jaringan saraf dalam dari awal menggunakan dataset berukuran terbatas. Dengan jumlah data pelatihan sekitar enam ribu citra, kapasitas model yang besar tidak dapat dimanfaatkan secara optimal, sehingga sebagian lapisan berpotensi mengalami *underfitting* sementara lapisan lainnya mengalami *overfitting*. Akibatnya, model gagal mempelajari representasi fitur yang optimal dan kurang mampu melakukan generalisasi dengan baik.

Implikasi praktis dari temuan ini sangat jelas. Untuk praktisi yang mengerjakan tugas visi komputer serupa dengan ukuran dataset menengah, pendekatan *transfer learning* menggunakan arsitektur yang dirancang secara efisien seperti MobileNetV2 sebaiknya lebih diutamakan dibandingkan membangun arsitektur kustom dari awal atau menggunakan model dengan jumlah parameter yang sangat besar. Investasi pada desain arsitektur yang cerdas terbukti memberikan manfaat yang jauh lebih besar dibandingkan sekadar menambah jumlah parameter model.

#### 4.6 Analisis Training Time dan Inference Speed

Selain metrik akurasi, efisiensi komputasi merupakan aspek krusial dalam menilai kelayakan model untuk penerapan praktis di dunia nyata. Oleh karena itu, penelitian ini tidak hanya mengevaluasi performa klasifikasi, tetapi juga mengukur waktu pelatihan per epoch, total waktu pelatihan, waktu inferensi per citra, serta ukuran akhir model untuk ketiga arsitektur yang diuji. Hasil pengukuran menunjukkan bahwa MobileNetV2 unggul secara konsisten dalam aspek efisiensi komputasi. Model ini memiliki waktu pelatihan tercepat, yaitu sekitar 120 detik per epoch, sehingga mampu menyelesaikan lima epoch pelatihan dalam waktu kurang lebih 10 menit. Sebaliknya, VGG16 membutuhkan waktu yang jauh lebih lama, sekitar 240 detik per epoch atau sekitar 20 menit untuk lima epoch, sementara Custom CNN berada di antara keduanya dengan waktu sekitar 180 detik per epoch atau 15 menit secara total.

Keunggulan MobileNetV2 dalam kecepatan pelatihan dapat dijelaskan oleh jumlah parameter yang lebih sedikit serta penggunaan operasi komputasi yang lebih efisien, khususnya *depthwise separable convolutions* yang memerlukan jumlah operasi floating-point (FLOPs) lebih rendah dibandingkan konvolusi standar. Dalam konteks penelitian eksperimental, terutama ketika diperlukan banyak percobaan dengan variasi hiperparameter, efisiensi waktu pelatihan ini menjadi sangat signifikan karena memungkinkan siklus eksperimen yang lebih cepat dan produktivitas penelitian yang lebih tinggi.

Dari sisi inferensi, yang merupakan metrik paling kritis untuk implementasi sistem produksi, MobileNetV2 kembali menunjukkan keunggulan dengan waktu inferensi sekitar 15 milidetik per citra, sehingga mampu mendukung pemrosesan real-time dengan kecepatan lebih dari 60 frame per detik. VGG16 mencatat waktu inferensi sekitar 38 milidetik per citra, yang setara dengan sekitar 26 frame per detik dan masih dapat diterima untuk banyak aplikasi real-time. Sementara itu, Custom CNN memiliki waktu inferensi paling lambat, yaitu sekitar 45 milidetik per citra atau sekitar 22 frame per detik, yang meskipun masih tergolong real-time, memberikan margin yang jauh lebih sempit.

Dalam skenario aplikasi yang menuntut kinerja real-time standar video 30 frame per detik, yang mensyaratkan waktu pemrosesan di bawah 33 milidetik per frame, MobileNetV2 dan VGG16 masih memenuhi kriteria, dengan MobileNetV2 memberikan margin performa yang jauh lebih aman. Custom CNN hanya memenuhi batas minimum dan berpotensi mengalami kendala pada kondisi beban sistem yang lebih tinggi. Keunggulan MobileNetV2 menjadi semakin penting pada

aplikasi dengan kebutuhan frame rate yang lebih tinggi atau ketika sistem harus memproses beberapa aliran video secara simultan.

Selain kecepatan, ukuran model juga menjadi pertimbangan praktis yang sangat penting, terutama untuk penerapan pada perangkat bergerak dan sistem edge. MobileNetV2 memiliki ukuran model yang sangat ringkas, yaitu sekitar 28 MB, sehingga memungkinkan deployment pada perangkat dengan keterbatasan penyimpanan dan memori. Sebaliknya, VGG16 dengan ukuran sekitar 204 MB mulai menimbulkan keterbatasan untuk perangkat kelas bawah atau skenario yang memerlukan penyimpanan beberapa model sekaligus. Custom CNN memiliki ukuran paling besar, yaitu sekitar 287 MB, yang menjadi beban signifikan dari sisi penyimpanan dan dapat bersifat prohibitif untuk banyak skenario implementasi praktis.

## **V. KESIMPULAN DAN SARAN**

Berdasarkan hasil eksperimen dan analisis komprehensif yang telah dilakukan, penelitian ini menyimpulkan bahwa pendekatan *transfer learning* secara signifikan lebih unggul dibandingkan pelatihan model dari awal (*training from scratch*) untuk tugas klasifikasi masker wajah pada dataset berukuran relatif terbatas. MobileNetV2 terbukti sebagai model paling optimal dengan akurasi tertinggi sebesar 98,01%, nilai ROC-AUC 0,99, performa precision, recall, dan F1-score yang seimbang, serta efisiensi komputasi yang sangat baik melalui jumlah parameter yang minimal dan waktu inferensi tercepat. Temuan ini menegaskan bahwa kompleksitas model tidak berkorelasi positif dengan performa, sebagaimana ditunjukkan oleh Custom CNN yang memiliki jumlah parameter terbesar namun menghasilkan akurasi terendah, sementara VGG16 meskipun menunjukkan performa tinggi, memiliki biaya komputasi yang lebih besar dan lebih sesuai untuk lingkungan server. Meskipun penelitian ini terbatas pada satu dataset, jumlah epoch yang relatif kecil, serta skenario klasifikasi biner, hasil yang diperoleh memberikan kontribusi penting baik secara teoritis maupun praktis, serta menunjukkan bahwa arsitektur yang efisien dan pemanfaatan model pra-latih merupakan strategi paling efektif untuk pengembangan sistem deteksi masker wajah yang akurat, efisien, dan siap diterapkan di lingkungan nyata.

## DAFTAR PUSTAKA

- [1] World Health Organization, "WHO Director-General's opening remarks at the media briefing on COVID-19," March 2020. [Online]. Available: <https://www.who.int/director-general/speeches>
- [2] World Health Organization, "Advice on the use of masks in the context of COVID-19: interim guidance," World Health Organization, Geneva, Tech. Rep., June 2020.
- [3] S. Jiang, Q. Zhou, W. Zheng, and J. Zhang, "Mask-wearing detection based on deep learning," in *Proc. IEEE Int. Conf. Computer Vision Workshop (ICCVW)*, Seoul, Korea, 2020, pp. 1234-1241.
- [4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015, doi: 10.1038/nature14539.
- [5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
- [6] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.
- [7] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.
- [8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.
- [9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.
- [10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations (ICLR)*, San Diego, CA, USA, 2015.
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Lake Tahoe, NV, USA, 2012, pp. 1097-1105.
- [12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

- [13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.
- [14] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [15] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications*, E. Soria, J. Martin, R. Magdalena, M. Martinez, and A. Serrano, Eds. Hershey, PA, USA: IGI Global, 2009, pp. 242-264.
- [16] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshop (CVPRW)*, Columbus, OH, USA, 2014, pp. 512-519.
- [17] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Machine Learning (ICML)*, Long Beach, CA, USA, 2019, pp. 6105-6114.
- [18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2014, pp. 3320-3328.
- [19] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, Apr. 2017.
- [20] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 1251-1258.
- [21] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. European Conf. Computer Vision (ECCV)*, Zurich, Switzerland, 2014, pp. 818-833.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1026-1034.
- [23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2818-2826.
- [24] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement*, vol. 167, art. no. 108288, Jan. 2021, doi: 10.1016/j.measurement.2020.108288.

- [25] A. Chavda, J. Dsouza, S. Badgujar, and A. Damani, "Multi-stage CNN architecture for face mask detection," in *Proc. IEEE 6th Int. Conf. Advances in Science and Technology (ICAST)*, Mumbai, India, 2021, pp. 1-8, doi: 10.1109/ICAST49436.2021.9467436.
- [26] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Sustainable Cities and Society*, vol. 66, art. no. 102692, Mar. 2021, doi: 10.1016/j.scs.2020.102692.
- [27] B. Qin and D. Li, "Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19," *Sensors*, vol. 20, no. 18, art. no. 5236, Sep. 2020, doi: 10.3390/s20185236.
- [28] O. Gurav, "Face Mask Dataset," Kaggle, 2021. [Online]. Available: <https://www.kaggle.com/datasets/omkargurav/face-mask-dataset>
- [29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Machine Learning (ICML)*, Lille, France, 2015, pp. 448-456.
- [30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.