Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

Analisis Komparatif Arsitektur Convolutional Neural Network untuk Klasifikasi Kualitas Cabai dengan Implementasi Perangkat Mobile

Nur Ikhsanudin*, Muhamad Akrom

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia
Email: [1,*]111202214795@mhs.dinus.ac.id, [2]m.akrom@dsn.dinus.ac.id
Email Penulis Korespondensi: 1111202214795@mhs.dinus.ac.id
Submitted: **18/02/2026**; Accepted: **05/03/2026**; Published**: 06/03/2026**

Abstrak−Proses penyortiran kualitas cabai secara manual rentan terhadap subjektivitas dan inkonsistensi penilaian antar penyortir, yang dapat menurunkan nilai jual produk. Penelitian ini melakukan analisis komparatif terhadap tiga arsitektur Convolutional Neural Network (CNN), yaitu Custom CNN, MobileNetV3-Small, dan EfficientNetV2-B0, untuk klasifikasi kualitas cabai secara biner (Good/Bad) menggunakan dataset primer 1.383 citra cabai (684 kelas Good, 699 kelas Bad) yang dikumpulkan menggunakan kamera *smartphone*. Kelas Good mencakup cabai dengan permukaan mulus, warna segar, dan bebas bercak busuk, sementara kelas Bad mencakup cabai yang menunjukkan tanda pembusukan, cacat fisik, atau deformasi. Evaluasi dilakukan berdasarkan akurasi, *precision*, *recall*, *F1-Score*, AUC, waktu inferensi, dan ukuran model pasca-kuantisasi. Hasil menunjukkan bahwa EfficientNetV2-B0 mencapai akurasi tertinggi sebesar 92,0% (precision 92,4%, recall 92,0%, F1-Score 92,0%, AUC 0,961), MobileNetV3-Small memperoleh akurasi 87,7% dengan latensi inferensi server terendah (2,39 ms), dan Custom CNN mencapai akurasi 87,3% dengan ukuran model paling kompak (118 KB pasca-kuantisasi). Ketiga model diintegrasikan ke dalam prototipe aplikasi Android berbasis Flutter sebagai *proof-of-concept*, yang menampilkan hasil klasifikasi (Good/Bad), *confidence score*, dan latensi inferensi dengan waktu respons *end-to-end* berkisar 80–120 ms pada perangkat Xiaomi 13T. Penelitian ini berkontribusi dalam menyediakan data empiris komparatif tiga arsitektur CNN pada domain kualitas cabai, disertai pembangunan dataset primer dan validasi teknis deployment model pada perangkat mobile. Hasil penelitian ini diharapkan menjadi acuan dalam pemilihan arsitektur CNN untuk pengembangan sistem klasifikasi kualitas cabai berbasis mobile, khususnya sebagai langkah awal menuju penerapan sortir sederhana berskala kecil di tingkat petani.

Kata Kunci: Klasifikasi Cabai; Deep Learning; Convolutional Neural Network; EfficientNetV2; MobileNetV3; Aplikasi Mobile

Abstract−Manual chili quality sorting is susceptible to subjectivity and inter-assessor inconsistency, which can reduce product market value. This study conducts a comparative analysis of three Convolutional Neural Network (CNN) architectures Custom CNN, MobileNetV3-Small, and EfficientNetV2-B0 for binary chili quality classification (Good/Bad) using a primary dataset of 1,383 chili images (684 Good-class, 699 Bad-class) captured with a smartphone camera. The Good class includes chili with a smooth surface, fresh color, and no decay spots, while the Bad class includes chili showing signs of decay, physical defects, or deformation. Evaluation was conducted based on accuracy, precision, recall, F1-Score, AUC, inference time, and post-quantization model size. The results show that EfficientNetV2-B0 achieved the highest accuracy of 92.0% (precision 92.4%, recall 92.0%, F1-Score 92.0%, AUC 0.961), MobileNetV3-Small obtained an accuracy of 87.7% with the lowest server-side inference latency (2.39 ms), and Custom CNN achieved 87.3% accuracy with the most compact model size (118 KB post-quantization). All three models were integrated into a Flutter-based Android application prototype as a proof-of-concept, displaying the classification result (Good/Bad), *confidence score*, and inference latency, with end-to-end response times ranging from 80 to 120 ms on a Xiaomi 13T device. This study contributes empirical comparative data on three CNN architectures in the chili quality classification domain, accompanied by the construction of a local dataset and technical validation of model deployment on a mobile device. The results of this study are expected to serve as a reference in selecting CNN architecture for the development of a mobile-based chili quality classification system, particularly as a first step toward the implementation of simple small-scale sorting at the farmer level.

Keywords: Chili Classification; Deep Learning; Convolutional Neural Network; EfficientNetV2; MobileNetV3; Mobile Application

1. PENDAHULUAN

Cabai (*Capsicum annuum*) merupakan salah satu komoditas hortikultura bernilai ekonomi tinggi yang memiliki peran strategis dalam perekonomian Indonesia. Sebagai negara agraris, Indonesia menempatkan cabai sebagai komoditas penting dengan nilai ekspor yang signifikan, khususnya cabai kering untuk pasar Eropa dan Asia [1]. Namun, sektor pertanian cabai masih menghadapi tantangan yang menghambat produktivitas dan efisiensi. Salah satu kendala utamanya adalah proses klasifikasi kualitas cabai yang mayoritas masih dilakukan secara manual. Metode ini sangat bergantung pada penilaian visual yang subjektif, sehingga rentan menghasilkan tingkat akurasi yang tidak konsisten [2], [3]. Pada proses *sorting* manual cabai, ditemukan bahwa ketergantungan pada faktor manusia seperti kelelahan, keterbatasan waktu, perbedaan persepsi kualitas antar penyortir, serta subjektivitas dalam menentukan kategori kualitas mengakibatkan hasil sortasi yang tidak konsisten dan dapat menurunkan nilai jual produk [4].

Kementerian Pertanian Indonesia telah menetapkan target ambisius untuk meningkatkan adopsi teknologi oleh petani mencapai 65-80% pada tahun 2020 dan 80-95% pada tahun 2024 sebagai bagian dari Rencana Strategis Kementerian Pertanian [5]. Target ini mengindikasikan urgensi transformasi digital dalam sektor pertanian untuk meningkatkan efisiensi, produktivitas, dan daya saing produk pertanian Indonesia di pasar global. Dalam konteks pertanian pintar, implementasi sistem klasifikasi otomatis berbasis *computer vision* dan *deep learning* menjadi solusi potensial untuk mengatasi keterbatasan metode manual [6]. Teknologi *Convolutional Neural Network* (CNN) telah menunjukkan performa superior dalam berbagai tugas klasifikasi citra dengan kemampuan ekstraksi fitur otomatis

Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

yang dapat mengenali pola kompleks dan variasi halus dalam data visual, termasuk dalam berbagai tugas klasifikasi di domain pertanian [7], [8], [9]. Namun, tantangan utama dalam implementasi CNN untuk aplikasi pertanian adalah kebutuhan akan sistem yang tidak hanya akurat dalam lingkungan laboratorium, tetapi juga efisien dan dapat dijalankan pada perangkat dengan sumber daya komputasi terbatas seperti *mobile device* [10].

Berbagai penelitian telah menunjukkan efektivitas CNN dalam klasifikasi kematangan dan kualitas komoditas pertanian, termasuk klasifikasi tingkat kematangan buah nanas menggunakan MobileNetV2 [11] serta tinjauan sistematis terhadap metode klasifikasi kematangan buah dan sayuran yang didominasi pendekatan CNN dan deep learning [12].

Dalam domain klasifikasi cabai secara khusus, beberapa penelitian telah mengeksplorasi berbagai pendekatan deep learning dengan fokus yang beragam, khususnya *Convolutional Neural Network* (CNN), telah menunjukkan keberhasilan yang signifikan dalam berbagai aplikasi klasifikasi citra, termasuk di bidang pertanian. Beberapa penelitian telah dilakukan dalam domain klasifikasi kualitas cabai menggunakan pendekatan *machine learning* dan *deep learning*. Herdiyeni dan Haristu [13] mengembangkan sistem klasifikasi kualitas cabai menggunakan metode YOLO untuk pengenalan citra secara *real-time* dengan akurasi 99,4% untuk *simple test* dan 75,6% untuk *challenging test*, menunjukkan potensi *deep learning* untuk klasifikasi *real-time*, namun belum mengeksplorasi arsitektur CNN yang lebih efisien untuk *deployment mobile* dan perbandingan dengan model lain.

Beberapa penelitian fokus pada pendekatan *hybrid* dan tingkat kematangan. Adiyantari et al. [4] mengimplementasikan metode *hybrid* CNN dan KNN untuk klasifikasi tingkat kematangan cabai rawit dengan akurasi 99,33%, namun pendekatan ini menambah kompleksitas sistem dan belum dioptimalkan untuk perangkat mobile. Khoiruddin dan Tena [14] melakukan klasifikasi buah dan sayuran termasuk cabai menggunakan MobileNetV2 dengan akurasi tinggi, namun penelitian bersifat umum dan tidak spesifik mengoptimalkan model untuk karakteristik cabai dengan dataset lokal Indonesia. Hanafi et al. [15] menggunakan *mobile inverted bottleneck convolutional* (MBConv) dari arsitektur EfficientNet untuk klasifikasi tahap kematangan cabai, menunjukkan potensi EfficientNet untuk efisiensi, namun fokus pada tingkat kematangan bukan kualitas komersial dan belum melakukan perbandingan sistematis dengan arsitektur lain.

Penelitian lain fokus pada deteksi penyakit tanaman cabai. Winiarti dan Khoirunnisa [16] mengembangkan aplikasi *mobile* untuk deteksi penyakit cabai menggunakan CNN dengan *transfer learning*, berhasil diimplementasikan pada *mobile Android*, namun fokus pada deteksi penyakit daun bukan klasifikasi kualitas buah. Setiono [17] mengklasifikasikan penyakit antraknosa pada cabai rawit dengan CNN membandingkan tiga *optimizer*, dimana Adam memberikan akurasi tertinggi 93,25%, menunjukkan pentingnya pemilihan *optimizer*, namun fokus pada penyakit bukan kualitas buah. Anggraeni et al. [3] mengembangkan sistem klasifikasi penyakit tanaman cabai menggunakan CNN dengan tiga kategori, namun belum melakukan perbandingan arsitektur untuk mengidentifikasi model optimal untuk *deployment* dengan keterbatasan sumber daya. Yudantoro et al. [18] mengevaluasi performa berbagai model *deep learning* untuk klasifikasi cabai kering, memberikan perbandingan komprehensif, namun fokus pada cabai kering dan belum mengevaluasi model yang dioptimalkan untuk *mobile deployment*. Di luar domain cabai, pendekatan serupa juga telah diterapkan pada komoditas hortikultura lain, pendekatan klasifikasi biner kualitas buah menggunakan CNN telah berhasil diterapkan pada buah lemon, di mana citra dikategorikan menjadi dua kelas kualitas baik dan buruk dengan evaluasi menggunakan *confusion matrix* , menunjukkan bahwa CNN mampu menjadi dasar sistem inspeksi kualitas produk hortikultura secara otomatis [19].

Berdasarkan kajian literatur, terdapat beberapa gap penelitian. Pertama, studi komparatif yang sistematis antara arsitektur CNN yang dirancang untuk efisiensi pada perangkat mobile seperti Custom CNN, MobileNetV3-Small, dan EfficientNetV2-B0 masih terbatas, khususnya dalam konteks klasifikasi kualitas cabai secara biner (Good/Bad) untuk mendukung keputusan penyaringan berbasis visi komputer. Kedua, meskipun beberapa penelitian menunjukkan potensi implementasi pada perangkat mobile, analisis performa komputasi yang komprehensif meliputi akurasi, waktu inferensi, dan ukuran model dalam kondisi *on-device* pada Android masih jarang dilakukan.

Penelitian ini bertujuan untuk melakukan evaluasi dan analisis komparatif terhadap tiga arsitektur CNN, Custom CNN, MobileNetV3-Small, dan EfficientNetV2-B0 dalam mengklasifikasikan kualitas cabai secara biner menggunakan dataset primer cabai. Analisis dilakukan berdasarkan tiga aspek utama: akurasi klasifikasi, waktu inferensi, dan ukuran model. Ketiga model diuji melalui prototipe aplikasi Android sebagai *proof-of-concept* untuk menilai performa *on-device*. Sebelum pengujian, model-model tersebut telah diquantize dan dikonversi ke format TensorFlow Lite (TFLite) guna memungkinkan inferensi langsung pada perangkat mobile. Penelitian ini belum banyak dilakukan secara menyeluruh pada literatur sebelumnya, khususnya dalam konteks perbandingan tiga arsitektur CNN untuk klasifikasi kualitas cabai rawit. Kebaruan penelitian ini terletak pada penggunaan dataset primer yang dikumpulkan langsung di lapangan dengan protokol multi-rotasi, serta dilakukannya validasi teknis deployment model pasca-kuantisasi pada perangkat Android melalui prototipe aplikasi berbasis Flutter sebagai *proof-of-concept.*

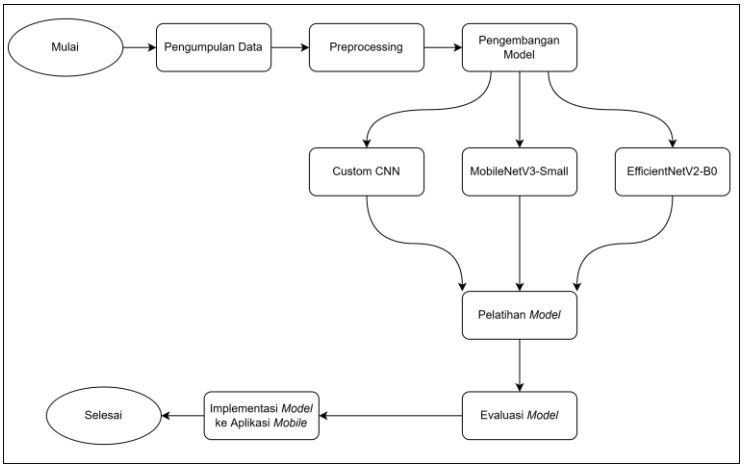
2. METODOLOGI PENELITIAN

1.
2.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

2.1 Tahapan Penelitian

Penelitian ini dilakukan melalui serangkaian tahapan sistematis dan menggunakan pendekatan eksperimental komparatif. Alur penelitian dimulai dari pengumpulan data citra, data *preprocessing*, perancangan arsitektur model, pelatihan dan validasi model, hingga tahap implementasi prototipe pada perangkat *mobile*. Rangkaian penelitian ditunjukkan pada Gambar 1 berikut:



Gambar 1. Rangkaian penelitian

2.2 Pengumpulan Data

Tahap awal penelitian ini adalah pengumpulan data primer yang dilakukan oleh penulis dengan bantuan petani di Kabupaten Boyolali. Akuisisi citra dilakukan menggunakan *smartphone* (Xiaomi 13T) dalam lingkungan pencahayaan terkontrol dengan latar belakang kertas HVS putih guna meminimalisir *noise* visual. Pada setiap pengambilan citra, objek difoto secara tunggal (satu buah cabai per citra) dan diposisikan di tengah bidang pandang. Untuk merepresentasikan variasi populasi, dataset dibangun dari berbagai sampel cabai rawit fisik yang berbeda. Setiap individu sampel kemudian difoto menggunakan protokol multi-rotasi (pengambilan gambar dari beberapa sudut pandang berbeda). Pendekatan ini berfungsi untuk meningkatkan keberagaman data orientasi secara alami tanpa menghilangkan detail morfologi asli objek. Melalui proses tersebut, terkumpul dataset primer sebanyak 1.383 citra yang diklasifikasikan ke dalam dua kelas biner mutu, yaitu kelas Good (684 citra) dan kelas Bad (699 citra). Berikut ini adalah contoh gambar dari kedua kelas tersebut:

Gambar 2. Perbandingan citra cabai (a) Bad dan (b) Good.

Pada Gambar 2 Citra (a) merupakan sampel yang dikategorikan Bad ditandai dengan adanya bercak kerusakan dan tekstur busuk pada permukaan kulit, sedangkan Citra (b) merupakan sampel Good yang menunjukkan kondisi cabai segar dengan permukaan kulit yang mulus dan tidak terdapat tanda kerusakan.

2.3 Pra-pemrosesan Data

Sebelum data dimasukkan ke dalam jaringan saraf tiruan, dilakukan proses *preprocessing*. Seluruh citra diubah ukurannya (*resize*) menjadi dimensi spasial standar 224 × 224 piksel (224 × 224 × 3), yang merupakan resolusi input standar untuk meningkatkan efisiensi *training* dan inferensi pada arsitektur CNN modern seperti MobileNetV3 [20] dan EfficientNetV2 [21]. Proses *resizing* ini memiliki dampak signifikan terhadap performa model dalam tugas klasifikasi [22]. Pada model custom CNN, proses penskalaan nilai piksel dilakukan secara manual dengan faktor 1/255 sehingga berada pada rentang 0–1, sedangkan untuk MobileNetV3-Small dan EfficientNetV2-B0 digunakan fungsi *preprocessing* bawaan dari masing-masing model. *Data splitting* dilakukan secara acak dengan *random seed* 42 untuk menjaga konsistensi. Dataset dibagi menjadi dua subset, yaitu *training set* sebanyak 80% dan *validation set* sebanyak 20%. Dalam penelitian ini, teknik augmentasi data tidak diterapkan pada fase pemuatan data (*data loading*).

Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

Variabilitas dataset dibangun melalui protokol pengambilan citra multi-rotasi untuk mencakup variasi orientasi secara alami. Transformasi berbasis warna secara khusus dihindari karena dikhawatirkan dapat mengaburkan ciri visual pembusukan seperti perubahan warna dan tekstur permukaan yang menjadi dasar pembedaan kelas Good dan Bad. Pemuatan data dilakukan dengan *batch size* sebesar 32.

2.4 Arsitektur Model

Penelitian ini mengimplementasikan tiga jenis arsitektur *deep learning* untuk mengklasifikasikan kualitas cabai: Custom CNN yang berfungsi sebagai baseline, MobileNetV3-Small, dan EfficientNetV2-B0. Ketiga arsitektur dipilih untuk membandingkan performa antara model yang dikembangkan secara mandiri dengan model *transfer learning* yang telah menunjukkan efektivitas tinggi dalam klasifikasi citra, terutama dari aspek kompleksitas arsitektur dan efisiensi komputasi.

a. Custom CNN (Baseline)
Model Custom CNN dirancang sebagai baseline untuk mengevaluasi performa arsitektur CNN sederhana yang dilatih dari awal (*trained from scratch*) dalam klasifikasi kualitas cabai. Berbeda dengan pendekatan transfer learning, model ini tidak memanfaatkan bobot *pre-trained*, sehingga memberikan perspektif terhadap seberapa efektif arsitektur konvolusional standar dapat mempelajari representasi fitur spesifik domain secara langsung dari dataset terbatas.
Arsitektur ini terdiri dari tiga blok konvolusi progresif dengan ukuran filter 3×3 dan fungsi aktivasi ReLU, masing-masing mengekstraksi 32, 64, dan 128 *feature maps* secara berturut-turut. Setiap blok konvolusi diikuti oleh operasi *max pooling* 2×2 untuk reduksi dimensi spasial dan ekstraksi fitur hierarkis dari *low-level* edges hingga *high-level semantic patterns*. Fitur yang diekstraksi kemudian diagregasi menggunakan *Global Average Pooling* untuk mengurangi *overfitting* dan kompleksitas komputasi. Pada tahap klasifikasi, *dense layer* dengan 128 *neuron* dan *dropout rate* 0.5 digunakan sebelum *output layer sigmoid* untuk prediksi probabilitas biner. Arsitektur lengkap ditampilkan pada Tabel 1.

Tabel 1. Arsitektur model Custom CNN

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18.496
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73.856
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 128)	0
dense (Dense)	(None, 128)	16.512
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129
Total params: 329,669 (1.26 MB)		
Trainable params: 109,889 (429.25 KB)		
Non-trainable params: 0 (0.00 B)		
Optimizer params: 219,780 (858.52 KB)		

Dengan total 329.669 parameter dan seluruh 109.889 parameter terlatih (tidak ada *frozen layers*), model ini menawarkan efisiensi komputasi yang signifikan dibandingkan arsitektur berbasis transfer learning. Implementasi baseline ini bertujuan untuk mengukur sejauh mana arsitektur konvolusi sederhana yang dikonstruksi dari awal dapat mengenali pola visual diskriminatif pada citra cabai tanpa memanfaatkan representasi fitur *pre-trained* dari dataset berskala besar, serta memberikan *benchmark* performa untuk mengevaluasi *added value* dari strategi *transfer learning*.

b. MobileNetV3-Small
Model MobileNetV3-Small [20] diimplementasikan dengan strategi *transfer learning*, yang memanfaatkan pengetahuan dari domain sumber untuk meningkatkan performa pada domain target dengan data terbatas [23]. MobileNetV3-Small merupakan hasil optimasi melalui *hardware-aware neural architecture search* yang secara eksplisit menargetkan efisiensi pada CPU perangkat *mobile*.
Arsitektur MobileNetV3-Small menggunakan *inverted residual structure* dengan *linear bottleneck* yang telah dioptimasi melalui algoritma NetAdapt untuk mengeliminasi layer yang tidak efisien. Inovasi kunci mencakup *integrasi squeeze-and-excitation (SE) modules* untuk *recalibration* fitur berbasis *channel attention*, penerapan *h-swish activation function* yang komputasional lebih efisien daripada *swish* standar, dan *redesign* pada layer-layer *expensive* di bagian *head network* untuk mereduksi latensi tanpa menurunkan akurasi signifikan. *Configuration head classification* yang sama diterapkan untuk memastikan fairness dalam evaluasi performa.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

Implementasi *transfer learning* memanfaatkan bobot *pre-trained* dari dataset ImageNet sebagai *feature extractor* dengan total 939.120 parameter *backbone* yang *di-freeze* (*non-trainable*). Output feature map dari *backbone* MobileNetV3-Small diproses melalui *layer Global Average Pooling* untuk agregasi spasial, menghasilkan vektor fitur yang kemudian diteruskan ke *classification head*. Arsitektur lengkap model ditampilkan pada Tabel 2, dengan total parameter 1.161.077, dimana hanya 6.4% parameter yang trainable untuk menjaga efisiensi komputasi.

Tabel 2. Arsitektur model MobileNetV3-Small

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 224, 224, 3)	0
MobileNetV3-Small (Functional)	(None, 7, 7, 576)	939,120
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 576)	0
dense_4 (Dense)	(None, 128)	73,856
dropout_2 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 1)	129
Total params: 1,161,077 (4.43 MB)		
Trainable params: 73,985 (289.00 KB)		
Non-trainable params: 939,120 (3.58 MB)		
Optimizer params: 147,972 (578.02 KB)		

MobileNetV3-Small merupakan arsitektur CNN yang dioptimasi untuk *deployment* pada perangkat dengan resource terbatas melalui penggunaan *depthwise separable convolution* dan *squeeze-and-excitation blocks*. Pada penelitian ini, model diinisialisasi dengan bobot pre-trained ImageNet sebagai dasar *transfer learning*. Seluruh lapisan backbone MobileNetV3-Small (939.120 parameter) dibekukan (*frozen*) untuk mempertahankan representasi fitur umum yang telah dipelajari, sementara hanya *classification head* yang dilatih ulang. *Classification head* terdiri dari *Global Average Pooling* untuk agregasi spasial, *dense layer* dengan 128 unit dan *dropout regularization* untuk mencegah *overfitting*, diakhiri dengan *single neuron output layer* dengan aktivasi sigmoid.

c. EfficientNetV2-B0

Model EfficientNetV2-B0 [21] diimplementasikan dengan pendekatan *transfer learning*. Model EfficientNetV2 menawarkan *trade-off* optimal antara akurasi dan efisiensi komputasi, mencapai ukuran model hingga 6.8x lebih kecil dibandingkan arsitektur *state-of-the-art* lainnya tanpa mengorbankan performa.

Implementasi *transfer learning* memanfaatkan bobot *pre-trained* EfficientNetV2-B0 dari dataset ImageNet sebagai *feature extractor* dengan total 5.919.312 parameter yang di-*freeze* (non-trainable). Layer-layer yang telah *di-pretrain* ini berfungsi mengekstrak fitur visual generik tingkat rendah hingga tingkat tinggi dari input gambar berukuran 224×224×3 piksel. Output dari *backbone* EfficientNetV2-B0 berupa *feature map* dengan dimensi 7×7×1280 selanjutnya diproses melalui layer *Global Average Pooling* untuk mereduksi dimensi spasial menjadi vektor fitur 1280-dimensi.

Untuk adaptasi pada tugas klasifikasi biner, ditambahkan *classification head* yang terdiri dari layer *Dense* dengan 128 *neuron* dan aktivasi ReLU, diikuti layer *Dropout* dengan rate 0.5 untuk regularisasi, dan layer *output Dense* dengan 1 *neuron* menggunakan aktivasi sigmoid untuk prediksi probabilitas biner. *Classification head* ini memiliki 164.097 parameter trainable. Arsitektur lengkap model ditunjukkan pada Tabel 3, dengan total parameter 6.411.605 (24.46 MB), dimana hanya 2.6% parameter yang dilatih ulang untuk menjaga efisiensi komputasi dan mencegah *overfitting*.

Tabel 3. Arsitektur model EfficientNetV2-B0

Layer (type)	Output Shape	Param #
input_layer_5 (InputLayer)	(None, 224, 224, 3)	0
efficientnetv2-b0 (Functional)	(None, 7, 7, 1280)	5,919,312
global_average_pooling2d_3 (GlobalAveragePooling2D)	(None, 1280)	0
dense_6 (Dense)	(None, 128)	163,968
dropout_3 (Dropout)	(None, 128)	0
dense_7 (Dense)	(None, 1)	129
Total params: 6,411,605 (24.46 MB)		
Trainable params: 164,097 (641.00 KB)		
Non-trainable params: 5,919,312 (22.58 MB)		
Optimizer params: 328,196 (1.25 MB)		

EfficientNetV2-B0 mengimplementasikan *compound scaling methodology* yang mengoptimalkan secara simultan *depth, width, dan resolution* jaringan *neural* untuk mencapai *state-of-the-art performance* dengan efisiensi

Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

parameter. Arsitektur ini mengintegrasikan *Fused-MBConv blocks* pada stage awal dan MBConv *blocks* pada stage lanjut, dilengkapi dengan *Progressive Learning strategy* yang secara adaptif menyesuaikan ukuran gambar dan regularisasi selama pelatihan. Kapasitas representasional yang tinggi dari EfficientNetV2-B0 diharapkan mampu mengekstraksi fitur visual kompleks dan subtle variations pada karakteristik kualitas cabai, berpotensi meningkatkan *discriminative power* dan generalisasi model pada dataset yang terbatas.

2.5 Konfigurasi Pelatihan

Ketiga model dilatih menggunakan konfigurasi *hyperparameter* yang sama, yaitu: *optimizer* Adam dengan *learning rate* 0.001, *binary crossentropy* sebagai fungsi *loss*, *batch size* 32, dan iterasi training sebanyak 10 *epoch*. Jumlah ini ditentukan berdasarkan observasi kurva validasi pada eksperimen awal, di mana *loss* validasi model transfer learning menunjukkan stabilisasi dalam rentang 8–10 *epoch*. Untuk Custom CNN, 10 *epoch* dipertahankan guna menjaga konsistensi evaluasi komparatif antar arsitektur.

2.6 Metrik Evaluasi

Pengukuran reliabilitas model hasil training dilakukan melalui metode evaluasi berbasis *Confusion Matrix*. Evaluasi diimplementasikan dengan cara membandingkan *output* prediksi model dengan label *ground truth* pada data validasi. Sesuai dengan kode pengujian yang diterapkan, metrik yang dikalkulasi meliputi Akurasi, *Precision*, *Recall*, *F1-Score,* dan *AUC*. Di samping itu, grafik visualisasi *training history plot* juga dibuat untuk mengobservasi stabilitas model dalam mengoptimalkan *loss* function dan akurasi selama proses iterasi *epoch*. Adapun rumus matematis yang digunakan untuk perhitungan metrik evaluasi adalah sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

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

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

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

$$AUC = \sum_{i=1}^{n-1} \frac{(FPR_{i+1}-FPR_i)(TPR_i-TPR_{i+1})}{2} \quad (5)$$

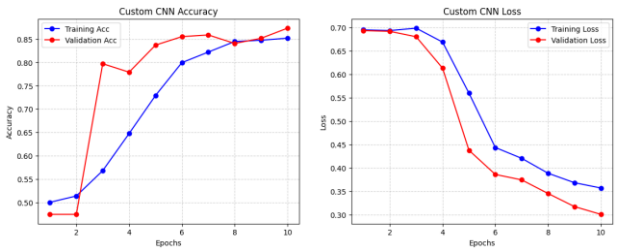
2.7 Perancangan Implementasi Mobile

Tahap *deployment* model ke aplikasi *mobile* melibatkan konversi dari format Keras ke TensorFlow Lite (.tflite) dengan optimasi *post-training quantization*. Teknik *quantization* mengkonversi representasi bobot model dari *floating-point* 32-bit (FP32) menjadi integer 8-bit (INT8), menghasilkan reduksi ukuran model hingga ~75% dan peningkatan *throughput* inferensi dengan memanfaatkan *integer arithmetic operations* yang lebih efisien pada CPU *mobile* [24]. Proses konversi TFLite juga mencakup optimasi *computational graph* melalui *operator fusion* dan *removal of redundant operations* untuk memaksimalkan efisiensi eksekusi. Prototipe aplikasi dikembangkan menggunakan *framework* Flutter dengan tflite_flutter plugin sebagai *inference engine*. Aplikasi dirancang dengan *pipeline* yang terdiri dari modul akuisisi gambar (kamera), *preprocessing* (*resize* 224×224 dan normalisasi *pixel values*), model *inference* melalui TFLite *interpreter*, dan user *interface* untuk menampilkan hasil prediksi berupa label kategori klasifikasi, *confidence score*, serta *inference latency* sebagai indikator performa.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pelatihan Model (Training Results)

a. Custom CNN (Baseline)



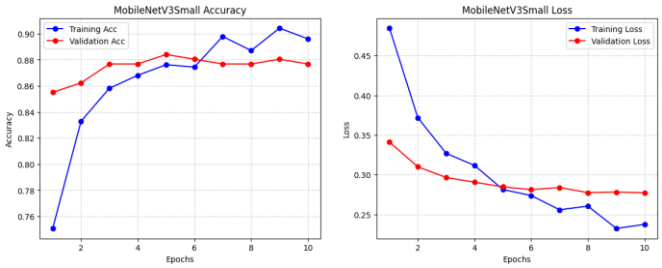
Gambar 3. Grafik Akurasi dan *Loss* Pelatihan Model Custom CNN

Model Custom CNN dikembangkan sebagai baseline untuk membandingkan efektivitas model *transfer learning*. Grafik akurasi pada Gambar 3 menunjukkan bahwa pada *epoch* awal, model masih memiliki performa mendekati

Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

acak dengan akurasi di sekitar 50%. Namun, mulai *epoch* ke-3 terjadi peningkatan akurasi validasi yang signifikan, menandakan bahwa model mulai mempelajari fitur visual yang relevan dari data citra. Seiring bertambahnya *epoch*, akurasi pelatihan dan validasi meningkat secara bertahap hingga mencapai akurasi validasi sebesar 87.3% pada *epoch* terakhir. Penurunan nilai *loss* yang konsisten pada data pelatihan dan validasi menunjukkan bahwa proses optimisasi berjalan dengan baik dalam lingkup distribusi dataset yang digunakan.

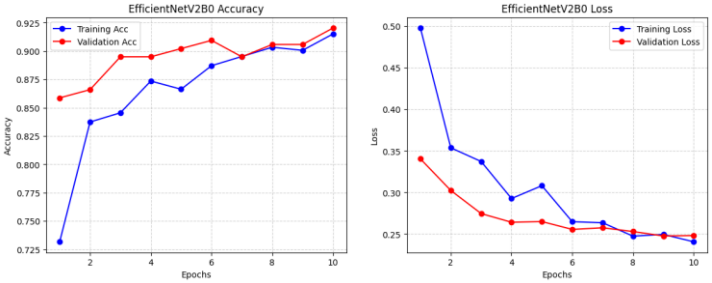
b. MobileNetV3-Small



Gambar 4. Grafik Akurasi dan *Loss* Pelatihan Model MobileNetV3-Small

Model MobileNetV3-Small pada Gambar 4 menunjukkan konvergensi yang jauh lebih cepat dibandingkan Custom CNN. Akurasi validasi sudah berada di atas 83% pada *epoch* pertama, yang mengindikasikan manfaat signifikan dari penggunaan *pretrained weights* dari ImageNet. Selama proses pelatihan, kurva akurasi dan *loss* relatif stabil dengan fluktuasi yang minimal. Akurasi validasi pada kisaran 87.7%, sedangkan nilai *loss* validasi bertahan di sekitar 0,28–0,30. Tidak terdapat perbedaan mencolok antara kurva akurasi pelatihan dan validasi, mengindikasikan konvergensi yang stabil dalam distribusi dataset yang digunakan.

c. EfficientNetV2-B0



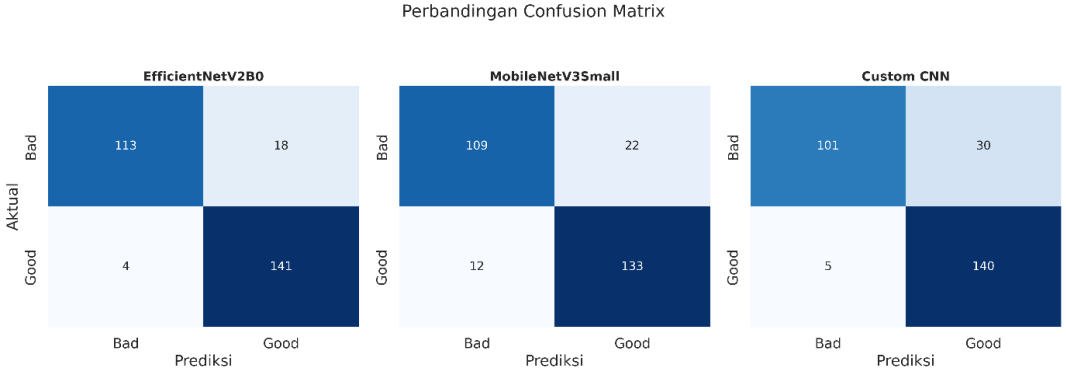
Gambar 5. Grafik Akurasi dan *Loss* Pelatihan Model EfficientNetV2-B0

EfficientNetV2-B0 pada Gambar 5 menunjukkan performa terbaik di antara seluruh model yang diuji. Grafik akurasi memperlihatkan peningkatan yang konsisten sejak *epoch* awal hingga *epoch* terakhir, dengan akurasi validasi mencapai 92.03%. Nilai *loss* validasi juga mengalami penurunan yang stabil dan berada pada kisaran 0,24–0,26. Meskipun terdapat fluktuasi kecil pada akurasi pelatihan, perbedaan antara kurva pelatihan dan validasi relatif kecil. *Divergensi* antara kurva training dan validasi yang relatif kecil mengindikasikan stabilitas konvergensi dalam distribusi dataset ini.

3.2 Evaluasi Performa Model

Untuk menilai lebih lanjut performa ketiga model, evaluasi dilakukan menggunakan *confusion matrix* dan metrik klasifikasi.

a. Evaluasi Menggunakan Confusion Matrix



Gambar 6. Hasil Confusion Matrix Setiap Model

Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

Confusion matrix digunakan untuk mengidentifikasi jumlah prediksi benar dan salah pada kelas good dan bad, sehingga dapat memberikan gambaran yang lebih rinci mengenai pola kesalahan model. Berdasarkan hasil yang diperoleh pada Gambar 6, EfficientNetV2-B0 menunjukkan performa terbaik dengan jumlah *true positive* dan *true negative* tertinggi serta kesalahan klasifikasi yang paling rendah. Model ini hanya menghasilkan sedikit kesalahan pada prediksi kelas good sebagai bad maupun sebaliknya. MobileNetV3-Small memiliki jumlah kesalahan yang sedikit lebih tinggi dibandingkan EfficientNetV2-B0, terutama pada kesalahan klasifikasi kelas bad menjadi good. Sementara itu, Custom CNN menunjukkan jumlah kesalahan klasifikasi terbesar, khususnya pada kelas bad, yang mengindikasikan keterbatasan model dalam mengekstraksi fitur visual yang lebih kompleks.

b. Evaluasi Metrik Klasifikasi

Evaluasi kinerja model dilakukan secara komprehensif menggunakan metrik klasifikasi standar, meliputi *accuracy*, *precision*, *Recall*, *F1-Score,* dan AUC. Evaluasi ini bertujuan untuk menilai kemampuan masing-masing model dalam mengklasifikasikan citra cabai ke dalam dua kelas, yaitu Good dan Bad, serta mengidentifikasi potensi kesalahan prediksi yang terjadi.

1. Hasil Evaluasi Custom CNN

Tabel 4 . Hasil Evaluasi Kinerja Custom CNN

Kelas	Precision	Recall	F1-Score	Support
Bad	0.95	0.77	0.85	131
Good	0.82	0.97	0.89	145
Accuracy			0.873	276
Macro Avg	0.89	0.87	0.87	276
Weighted Avg	0.88	0.87	0.87	276

Model Custom CNN pada Tabel 4 memperoleh nilai akurasi sebesar 87.3%. Meskipun performanya relatif mendekati MobileNetV3 Small, terdapat ketidakseimbangan yang cukup jelas pada beberapa metrik evaluasi. Nilai *precision* yang tinggi pada kelas bad (0.95) menunjukkan bahwa prediksi cabai buruk yang dihasilkan model sebagian besar benar. Namun, nilai *Recall* yang lebih rendah pada kelas tersebut (0.77) mengindikasikan bahwa cukup banyak cabai berkualitas buruk yang tidak terdeteksi dan salah diklasifikasikan sebagai cabai baik. Kondisi ini dapat menjadi kelemahan apabila model diterapkan pada sistem penyortiran otomatis. Pada kelas good, nilai *Recall* yang sangat tinggi (0.97) menunjukkan bahwa model cenderung lebih sensitif terhadap cabai berkualitas baik. Nilai *F1-Score* yang lebih rendah dibandingkan model lain mencerminkan keterbatasan Custom CNN dalam menjaga keseimbangan antara *precision* dan *Recall*.

2. Hasil Evaluasi MobileNetV3-Small

Tabel 5. Hasil Evaluasi Kinerja MobileNetV3-Small

Kelas	Precision	Recall	F1-Score	*Support*
Bad	0.90	0.83	0.87	131
Good	0.86	0.92	0.89	145
Accuracy			0.877	276
Macro Avg	0.88	0.87	0.88	276
Weighted Avg	0.88	0.88	0.88	276

MobileNetV3-Small pada Tabel 5 menghasilkan performa yang cukup kompetitif dengan akurasi sebesar 87.7%, disertai dengan keseimbangan metrik evaluasi yang relatif baik. Nilai *precision* pada kelas bad sebesar 0.90 menunjukkan bahwa sebagian besar prediksi cabai buruk yang dihasilkan model cukup akurat. Namun, nilai *Recall* pada kelas tersebut sebesar 0.83 mengindikasikan bahwa masih terdapat sejumlah cabai berkualitas buruk yang belum berhasil teridentifikasi. Pada kelas good, model menunjukkan *Recall* yang tinggi (0.92), yang berarti sebagian besar cabai berkualitas baik dapat dikenali dengan benar. Nilai *F1-Score* yang berada pada kisaran 0.87–0.89 menunjukkan bahwa MobileNetV3 Small memiliki performa klasifikasi yang cukup stabil, meskipun tidak sebaik EfficientNetV2-B0. Kinerja ini menunjukkan adanya kompromi antara akurasi dan efisiensi komputasi.

3. Hasil Evaluasi EfficientNetV2-B0

Tabel 6. Hasil Evaluasi Kinerja EfficientNetV2-B0

Kelas	Precision	Recall	F1-Score	Support
Bad	0.97	0.86	0.91	131
Good	0.89	0.97	0.93	145
Accuracy			0.920	276
Macro Avg	0.93	0.92	0.92	276
Weighted Avg	0.92	0.92	0.92	276

Berdasarkan hasil pada Tabel 6, EfficientNetV2-B0 menunjukkan performa paling unggul dibandingkan model lainnya. Model ini mampu mencapai nilai akurasi sebesar 92%, dengan keseimbangan yang baik antara

Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

precision dan *Recall* pada kedua kelas. Secara lebih rinci, nilai *precision* yang tinggi pada kelas bad (0.97) menunjukkan bahwa sebagian besar prediksi cabai berkualitas buruk yang dihasilkan model adalah benar. Hal ini penting untuk meminimalkan kesalahan klasifikasi cabai baik sebagai cabai buruk. Sementara itu, nilai *Recall* sebesar 0.86 pada kelas bad mengindikasikan bahwa mayoritas cabai berkualitas rendah berhasil terdeteksi dengan baik oleh model. Pada kelas good, nilai *Recall* yang sangat tinggi (0.97) menandakan kemampuan model dalam mengenali cabai berkualitas baik secara optimal. Nilai *F1-Score* yang tinggi dan seimbang pada kedua kelas menunjukkan bahwa EfficientNetV2-B0 memiliki stabilitas klasifikasi yang baik tanpa kecenderungan bias terhadap salah satu kelas.

c. Perbandingan Kinerja Antar Model
Untuk memberikan gambaran komparatif yang lebih jelas, Tabel 7 menyajikan ringkasan kinerja ketiga model berdasarkan beberapa metrik evaluasi utama, termasuk *Area Under Curve* (AUC) dan waktu inferensi.

Tabel 7. Perbandingan Metrik Evaluasi Model

Model	Accuracy	Precision	Recall	F1-Score	AUC	Inference (ms)
EfficientNetV2-B0	0.920	0.924	0.920	0.920	0.961	9.24
MobileNetV3-Small	0.877	0.878	0.877	0.876	0.953	2.39
Custom CNN	0.873	0.884	0.873	0.872	0.951	2.73

Hasil evaluasi menunjukkan kesesuaian performa masing-masing model dengan karakteristik arsitekturnya. EfficientNetV2-B0 mencapai akurasi tertinggi sebesar 92.0%, sedangkan MobileNetV3-Small memperoleh akurasi 87.7% dengan latensi inferensi terendah (2.39 ms) dan Custom CNN baseline mencapai akurasi 87.3%. Nilai *inference latency* pada tabel 7 diukur di lingkungan GPU server (Google Colab).

3.3 Ukuran Model

Ukuran model merupakan faktor penting dalam *deployment* aplikasi mobile, terutama untuk memastikan aplikasi dapat diunduh dan dijalankan dengan efisien pada perangkat dengan storage terbatas. Dalam penelitian ini, setiap model dikonversi ke dalam format TensorFlow Lite (TFLite) dengan dua varian: *non-quantized* (float32) dan *quantized* (int8).

Tabel 8. Perbandingan Ukuran Model

Model	*Non-quantized* (Float32)	Quantized (Int8)
EfficientNetV2-B0	23.514 KB	6.607 KB
MobileNetV3-Small	3.948 KB	1.171 KB
Custom CNN	434 KB	118 KB

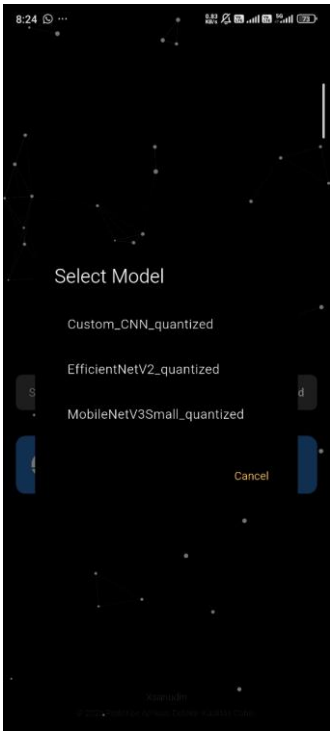
Berdasarkan Tabel 8 di atas, model Custom CNN memiliki ukuran paling kecil dengan 434 KB untuk versi *non-quantized* dan 118 KB untuk versi *quantized*. MobileNet menunjukkan ukuran yang cukup efisien dengan 3,948 KB (3.85 MB) yang dapat direduksi hingga 1,171 KB (1.14 MB) setelah *quantization*. Sementara itu, EfficientNet memiliki ukuran paling besar yaitu 23,514 KB (22.96 MB) untuk versi *non-quantized*, namun masih dapat direduksi menjadi 6,607 KB (6.45 MB) setelah proses *quantization*. Ketiga model menunjukkan reduksi ukuran yang konsisten sekitar 70-73% melalui proses *quantization*, sejalan dengan estimasi teoritis reduksi hingga 75% yang dilaporkan pada studi sebelumnya [25], mengkonfirmasi potensi efektivitas teknik ini dalam mengoptimalkan *deployment* model pada perangkat mobile dengan tetap mempertahankan performa prediksi yang baik.

3.4 Implementasi dan Pengujian pada Aplikasi Mobile

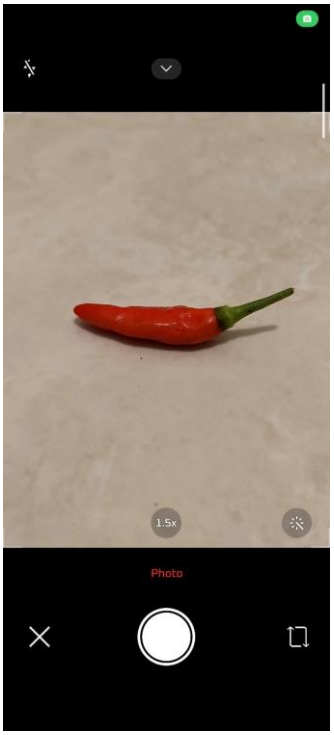
Model klasifikasi diterapkan dalam bentuk prototipe aplikasi mobile untuk mengevaluasi kemampuannya dalam penggunaan di perangkat bergerak sebagai proof-of-concept, bukan sebagai produk siap pakai. Prototipe dikembangkan menggunakan *framework* Flutter dan terdiri dari empat komponen utama: antarmuka pengguna, modul praproses citra, modul inferensi model, dan modul penampil hasil. Antarmuka dirancang sesederhana mungkin agar memudahkan pengambilan gambar dan menampilkan hasil prediksi kelas cabai (Tidak Layak sebagai Bad pada Gambar 11 dan Layak sebagai Good pada Gambar 10). Alur kerja aplikasi dimulai saat pengguna mengambil foto cabai melalui kamera *smartphone*. Gambar yang diperoleh kemudian melewati tahap praproses, yaitu *resize* ke dimensi 224×224 piksel, normalisasi nilai piksel, dan penyesuaian format tensor sesuai kebutuhan masing-masing model. Selanjutnya, inferensi dijalankan menggunakan TFLite interpreter pada CPU perangkat tanpa aktivasi GPU delegate maupun NNAPI. Hasil akhir yang ditampilkan kepada pengguna mencakup label klasifikasi (Good/Bad), *confidence score*, dan latensi inferensi. Ketiga model diintegrasikan dan diuji pada perangkat Xiaomi 13T seperti pada Gambar 9. Waktu respons *end-to-end*, yang mencakup keseluruhan alur dari praproses hingga tampilnya hasil di layar, untuk ketiga model berkisar 80–120 ms. Nilai ini berbeda dengan latensi inferensi model murni yang diukur di lingkungan GPU server (2,39–9,24 ms), sehingga keduanya perlu dibedakan dalam interpretasi performa. Prototipe aplikasi terlihat pada Gambar 7 hingga Gambar 11 berikut:

Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

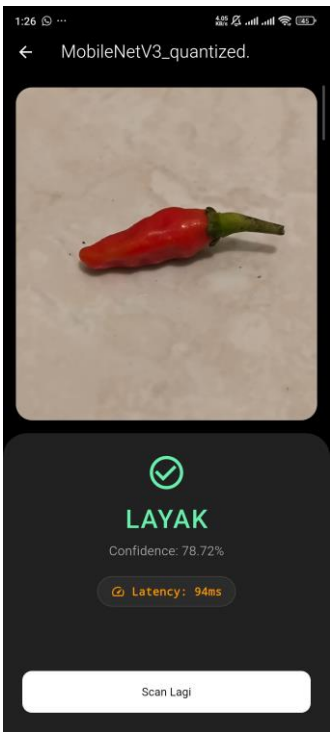
Gambar 7. Dashboard Pengguna



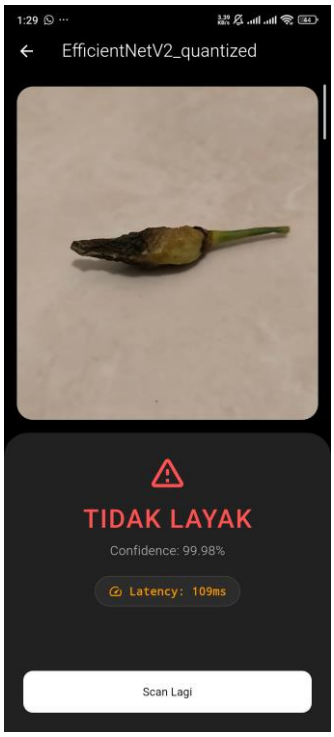
Gambar 8. Menu Pemilihan Model



Gambar 9. Halaman Tangkap Citra



Gambar 10. Prediksi Layak (good)



Gambar 11. Prediksi tidak layak (bad).

4. KESIMPULAN

Penelitian ini melakukan analisis komparatif terhadap tiga arsitektur CNN Custom CNN, MobileNetV3-Small, dan EfficientNetV2-B0 untuk klasifikasi kualitas cabai secara biner menggunakan dataset primer 1.383 citra. Custom CNN digunakan sebagai *baseline* tanpa *pretrained weights*, MobileNetV3-Small dipilih karena dirancang untuk efisiensi inferensi pada CPU perangkat mobile, dan EfficientNetV2-B0 dievaluasi untuk menguji kapasitas *transfer learning* berbasis *compound scaling*. Hasil evaluasi menunjukkan bahwa EfficientNetV2-B0 mencapai akurasi tertinggi sebesar 92,0% (*precision* 92,4%, *recall* 92,0%, *F1-Score* 92,0%, AUC 0,961), MobileNetV3-Small memperoleh akurasi 87,7% dengan latensi inferensi server terendah (2,39 ms), dan Custom CNN mencapai akurasi

Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

87,3% dengan ukuran model paling kompak (118 KB pasca-kuantisasi). Latensi 2,39–9,24 ms diukur di lingkungan GPU server, sedangkan waktu respons *end-to-end* pada perangkat Xiaomi 13T melalui eksekusi CPU berkisar 80–120 ms. Ketiga model dikonversi ke format TFLite dengan kuantisasi INT8 dan diintegrasikan ke dalam prototipe aplikasi Android berbasis Flutter yang menampilkan hasil klasifikasi (Good/Bad), *confidence score*, dan latensi inferensi. Pengujian pada perangkat Xiaomi 13T mengkonfirmasi bahwa ketiga model dapat berjalan on-device sebagai *proof-of-concept* pada perangkat *mobile*. Berdasarkan hasil pengujian pada dataset yang digunakan, EfficientNetV2-B0 mencapai akurasi tertinggi, sementara MobileNetV3-Small menawarkan latensi inferensi server terendah (2,39 ms) dan ukuran model pasca-kuantisasi yang lebih kompak (1.171 KB), sehingga menjadi pertimbangan relevan untuk deployment dengan keterbatasan storage. Model Custom CNN dalam penelitian ini memiliki ukuran yang paling kecil dan kompak dibanding model lainnya. Hasil penelitian ini diharapkan dapat menjadi acuan dalam pemilihan arsitektur CNN untuk pengembangan sistem klasifikasi kualitas cabai berbasis *mobile*, khususnya sebagai langkah awal menuju penerapan sortir sederhana berskala kecil di tingkat petani. Keterbatasan utama adalah penelitian ini dirancang untuk skenario satu citra per input, sehingga penggunaan pada sistem multi-objek atau alur sortir skala kecil berkelanjutan perlu dilengkapi dengan modul deteksi objek sebelum klasifikasi. Oleh karena itu, pengembangan selanjutnya dapat mengeksplorasi integrasi modul deteksi dan klasifikasi secara bersamaan, serta pengujian pada alur sortir nyata untuk menilai performa dan kestabilan sistem dalam kondisi operasional yang sesungguhnya.

REFERENCES

[1] Y. A. Azis, N. Khuriyati, and A. Suyantohadi, "Classification of Dried Chilli Quality Using Image Processing," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 686, no. 1, p. 012058, Mar. 2021, doi: 10.1088/1755-1315/686/1/012058.

[2] M. A. Aziz, W. M. Arif Mohamad Nazir, A. M. Ali, and J. Abawajy, "Chili Ripeness Grading Simulation Using Machine Learning Approach," *in Proc. IEEE Int. Conf. Comput. (ICOCO),* pp. 253–258, 2021, doi: 10.1109/ICOCO53166.2021.9673572.

[3] D. S. Anggraeni, A. Widayana, P. D. Rahayu, and C. Rozikin, "Metode Algoritma Convolutional Neural Network pada Klasifikasi Penyakit Tanaman Cabai," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 7, no. 1, pp. 73–78, Aug. 2022, doi: 10.30998/string.v7i1.13304.

[4] Y. Adiyantari, M. Rifki, B. Ulum, B. Rahmat, and M. H. P. Swari, "Implementasi Metode CNN Dan K-Nearest Neighbor Untuk Klasifikasi Tingkat Kematangan Tanaman Cabai Rawit," *Modem : Jurnal Informatika dan Sains Teknologi*, vol. 2, no. 3, pp. 112–123, Jul. 2024, doi: 10.62951/modem.v2i3.131.

[5] U. Mahdiyah, L. S. Wahyuniar, and S. Rochana, "Klasifikasi Kualitas Citra Cabai Dengan Menggunakan Algoritma Gradien Boosting," *JAMI: Jurnal Ahli Muda Indonesia*, vol. 4, no. 1, pp. 58–66, Jun. 2023, doi: 10.46510/jami.v4i1.137.

[6] N. Arifin, C. N. Insani, M. Milasari, and M. F. Rasyid, "Horticulture Smart Farming for Enhanced Efficiency in Industry 4.0 Performance," *Jurnal Teknik Informatika (Jutif)*, vol. 5, no. 5, pp. 1405–1412, Oct. 2024, doi: 10.52436/1.jutif.2024.5.5.2728.

[7] D. Li, C. Zhang, J. Li, M. Li, M. Huang, and Y. Tang, "MCCM: Multi-Scale Feature Extraction Network for Disease Classification and Recognition of Chili Leaves," *Front. Plant Sci.*, vol. 15, p. 1367738, May 2024, doi: 10.3389/fpls.2024.1367738.

[8] S. Rahman, R. A. Setyadi, A. Indrawati, A. Sembiring, and M. Zen, "Improving the Accuracy of Chili Leaf Disease Classification with ResNet and Fine-Tuning Strategy," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 10, pp. 247–255, Oct. 2024, doi: 10.14569/IJACSA.2024.0151027.

[9] D. Bhatt *et al.*, "CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope," *Electronics*, vol. 10, no. 20, p. 2470, Oct. 2021, doi: 10.3390/electronics10202470.

[10] S. Tai, Z. Tang, B. Li, S. Wang, and X. Guo, "Intelligent Recognition and Automated Production of Chili Peppers: A Review Addressing Varietal Diversity and Technological Requirements," *Agriculture*, vol. 15, no. 11, p. 1200, May 2025, doi: 10.3390/agriculture15111200.

[11] D. Liandaputra and A. Zahra, "Classification of Pineapple (Ananas comosus l.) Maturity Level Using Deep Learning Method," *G-Tech: Jurnal Teknologi Terapan*, vol. 8, no. 2, pp. 1091–1103, Apr. 2024, doi: 10.33379/gtech.v8i2.4122.

[12] R. Saktriawindarta and K. Kusrini, "Methods for Classifying Fruit and Vegetable Ripe Levels: A Systematic Review," *G-Tech: Jurnal Teknologi Terapan*, vol. 8, no. 4, pp. 2344–2354, Oct. 2024, doi: 10.70609/gtech.v8i4.5067.

[13] Sudianto, Y. Herdiyeni, A. Haristu, and M. Hardhienata, "Chilli Quality Classification Using Deep Learning," *in Proc. 2020 Int. Conf. Comput. Sci. Appl. Agric. (ICOSICA)*, Sep. 2020, doi: 10.1109/ICOSICA49951.2020.9243176.

[14] M. Khoiruddin and S. Tena, "Fruit and Vegetable Classification Using Convolutional Neural Network with MobileNetV2," *J. Appl. Res. Comput. Sci. Inf. Syst.*, vol. 2, no. 2, pp. 203–210, Dec. 2024, doi: 10.61098/jarcis.v2i2.197.

[15] Z. Ibrahim, M. Hanafi, S. M. Shafie, and S. M. Syed Ahmad, "Classification of C. annuum and C. frutescens Ripening Stages: How Well Does Deep Learning Perform?," *IIUM Engineering Journal*, vol. 25, no. 2, pp. 167–178, Jul. 2024, doi: 10.31436/iiumej.v25i2.2769.

[16] S. Winiarti and I. I. Khoirunnisa, "Mobile Application Development for Chili Disease Detection with Convolutional Neural Network," *International Journal of Informatics and Computation*, vol. 6, no. 2, pp. 97–112, Dec. 2024, doi: 10.35842/ijicom.v6i2.93.

[17] M. Setiono, "Klasifikasi Penyakit Antraknosa Citra Cabai Rawit Dengan Metode Convolutional Neural Network (CNN)," *JATISI*, vol. 11, no. 2, pp. 308–320, Jun. 2024, doi: 10.35957/jatisi.v11i2.8039.

[18] T. R. Yudantoro, M. Djaeni, R. R. Isnanto, and Prayitno, "Deep Learning Models Performance for Classifying Dried Chili Based on Digital Image Analysis," *JOIV : International Journal on Informatics Visualization*, vol. 9, no. 5, pp. 1951–1963, Sep. 2025, doi: 10.62527/joiv.9.5.2919.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 4, March 2026 Page: 2439−2450
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9419

[19] S. Mahdiyyah *et al.*, "Implementasi Object Detection untuk Deteksi Kualitas pada Buah Lemon dengan CNN," *Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI)*, vol. 6, no. 1, pp. 215–223, Jan. 2025, doi: 10.30998/jrami.v6i01.13417.
[20] A. Howard *et al.*, "Searching for MobileNetV3," *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 1314–1324, Oct. 2019, doi: 10.1109/ICCV.2019.00140.
[21] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," *in Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 139, pp. 10096–10106, Jul. 2021, doi: 10.48550/arXiv.2104.00298.
[22] H. Talebi and P. Milanfar, "Learning to Resize Images for Computer Vision Tasks," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 487–496, Oct. 2021, doi: 10.1109/ICCV48922.2021.00055.
[23] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021, doi: 10.1109/JPROC.2020.3004555.
[24] B. Jacob *et al.*, "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2704–2713, Dec. 2018, doi: 10.1109/CVPR.2018.00286.
[25] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A Survey of Quantization Methods for Efficient Neural Network Inference," *Low-Power Computer Vision*, pp. 291–326, Jan. 2022, doi: 10.1201/9781003162810-13.