



Perbandingan Performa Custom CNN, MobileNetV2, dan InceptionV3 untuk Identifikasi Penyakit Daun Tanaman

Krismono Sadi*, Moh Noor Al Azam

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Narotama, Surabaya, Indonesia

Email: ¹*blasterdark300@gmail.com, ²noor.azam@narotama.ac.id

Email Penulis Korespondensi: blasterdark300@gmail.com

Abstrak—Sensitivitas arsitektur *deep learning* terhadap ukuran *batch* (*batch size*) kerap diasumsikan seragam antar model, padahal penelitian ini menemukan pola yang justru berlawanan arah antara model yang dilatih dari awal (*from scratch*) dan model *Transfer Learning*. Menggunakan dataset multikomoditas berskala besar (72.000 citra, 72 kelas penyakit dari 21 komoditas), Custom CNN menunjukkan performa yang terus meningkat seiring pembesaran *batch size*, sedangkan MobileNetV2 dan InceptionV3 yang mengandalkan bobot pra-latih ImageNet melalui strategi *Feature Extraction* dan *Fine-Tuning* dua fase justru mencapai performa terbaik pada *batch size* terkecil (8). Pola ini konsisten dengan kerangka teoretis *sharp-flat minima*, dan signifikansinya diuji secara statistik menggunakan uji proporsi dua sampel ($p < 0.0001$ pada seluruh perbandingan antar *batch size* Custom CNN). MobileNetV2 pada konfigurasi optimalnya unggul signifikan dibandingkan InceptionV3 (*test accuracy* 97,57% berbanding 96,35%; $z = 4,24$, $p < 0,0001$) sekaligus jauh lebih ringkas ($\pm 2,96$ juta parameter, sekitar 8 kali lebih kecil). Analisis *Explainable AI* (Confusion Matrix, t-SNE, dan Grad-CAM) mengonfirmasi bahwa keunggulan ini bersumber dari representasi fitur yang lebih terpisah antar kelas dan atensi model yang terkonsentrasi pada region lesi penyakit bukan artefak dataset sehingga MobileNetV2 pada konfigurasi *batch size* kecil layak dipertimbangkan sebagai kandidat *deployment* perangkat mobile untuk mendukung pertanian cerdas (*smart farming*).

Kata Kunci: Deep Learning; Identifikasi Penyakit Daun; Transfer Learning; MobileNetV2; Batch Size; Explainable AI

Abstract—Batch-size sensitivity in deep learning architectures is often assumed to behave uniformly, yet this study finds an opposite-direction pattern between architectures trained from scratch and those relying on Transfer Learning. Using a large-scale multi-commodity dataset (72,000 images, 72 disease classes from 21 commodities), Custom CNN improves consistently as batch size grows, whereas MobileNetV2 and InceptionV3 built on ImageNet pretrained weights via two-phase Feature Extraction and Fine-Tuning achieve their best performance at the smallest batch size (8), consistent with the sharp-versus-flat minima theoretical framework. The pattern is confirmed statistically using two-proportion z-tests ($p < 0.0001$ across all Custom CNN batch-size comparisons). At its optimal configuration, MobileNetV2 significantly outperforms InceptionV3 (*test accuracy* 97.57% vs. 96.35%; $z = 4.24$, $p < 0.0001$) while being far more compact (± 2.96 million parameters, roughly 8 times smaller). Explainable AI analysis (Confusion Matrix, t-SNE, and Grad-CAM) confirms this advantage stems from more separable class-level feature representations and attention concentrated on lesion regions rather than dataset artifacts, making MobileNetV2 at a small batch size a strong candidate for mobile deployment in smart farming.

Keywords: Deep Learning; Plant Disease Identification; Transfer Learning; MobileNetV2; Batch Size; Explainable AI

1. PENDAHULUAN

Kesehatan tanaman adalah kunci produksi pertanian, namun serangan penyakit pada daun sering kali merusak hasil panen secara signifikan. Hingga kini, identifikasi penyakit masih mengandalkan pengamatan manual yang bersifat subjektif, lambat, dan terbatas oleh jumlah tenaga ahli di lapangan. Permasalahan ini menjadi semakin kompleks pada konteks pertanian dengan keragaman komoditas yang tinggi seperti di Indonesia, karena setiap komoditas mulai dari tanaman hortikultura seperti tomat, terong, dan kentang, hingga tanaman perkebunan seperti kopi dan tebu memiliki jenis penyakit daun yang berbeda dengan gejala visual yang kerap tumpang tindih antar kelas, sehingga sulit dibedakan hanya melalui pengamatan mata telanjang. Keterlambatan diagnosis pada skala lapangan berpotensi menyebabkan penyebaran penyakit meluas sebelum tindakan penanganan dilakukan, yang pada akhirnya berdampak langsung pada penurunan produktivitas dan kerugian ekonomi petani. Penerapan teknologi *deep learning* melalui *convolutional neural network* (CNN) menjadi solusi efektif untuk otomatisasi diagnosis visual karena mampu mempelajari pola tekstur dan bentuk lesi penyakit secara langsung dari citra tanpa memerlukan rekayasa fitur manual. Namun, variasi tekstur daun dan morfologi penyakit pada skala dataset besar yakni 72.000 citra dengan 72 kelas penyakit dari 21 komoditas tanaman memerlukan arsitektur model dengan presisi tinggi serta strategi pelatihan yang tepat agar mampu menggeneralisasi dengan baik pada seluruh kelas, termasuk kelas dengan kemiripan visual tinggi maupun kelas dengan jumlah sampel yang relatif lebih sedikit.

Untuk mengatasi keterbatasan tersebut, penelitian ini membandingkan tiga pendekatan CNN yang merepresentasikan filosofi pelatihan berbeda sebagai *metode pembandingan*: Custom CNN yang dirancang dan dilatih dari awal (*from scratch*) tanpa bergantung pada bobot pra-latih, serta MobileNetV2 dan InceptionV3 yang memanfaatkan *Transfer Learning* dari bobot ImageNet melalui skema dua fase (*Feature Extraction* dan *Fine-Tuning*). Ketiga arsitektur ini dipilih karena mewakili tiga titik berbeda dalam spektrum kompleksitas dan strategi pelatihan CNN yang umum digunakan pada identifikasi penyakit tanaman: arsitektur ringkas hasil rancangan sendiri (Custom CNN), arsitektur ringkas pra-latih (MobileNetV2), dan arsitektur dalam pra-latih (InceptionV3), sehingga perbandingan ketiganya memungkinkan analisis *trade-off* antara ukuran model, strategi pelatihan, dan performa secara lebih komprehensif dibandingkan penelitian yang hanya membandingkan varian dalam satu keluarga arsitektur.

Pemilihan ketiga arsitektur tersebut didasarkan pada peta posisi penelitian terdahulu sebagai berikut. Penelitian terdahulu telah menunjukkan potensi CNN dalam identifikasi penyakit tanaman. Mohanty dkk. (2016) [1] menggunakan dataset PlantVillage dan menunjukkan akurasi CNN lebih dari 99%, menjadi dasar kuat penggunaan *deep learning* dalam



identifikasi penyakit tanaman, meskipun performa setinggi itu diperoleh pada citra berlatar belakang seragam hasil akuisisi laboratorium sehingga berpotensi menurun tajam pada kondisi lapangan yang lebih bervariasi. Palupiningsih, Sujiwanto, dan Prawirodirjo (2023) membandingkan VGG, MobileNet, dan InceptionV3 pada klasifikasi kesehatan daun tomat dan menemukan InceptionV3 memberikan akurasi tertinggi (98%) [2], diikuti VGG (93%) dan MobileNet (88%), namun tidak mengevaluasi sensitivitas ketiga arsitektur tersebut terhadap variasi *batch size* maupun skala kelas penyakit yang lebih luas. Karno dkk. (2022) mengoptimalkan EfficientNetB3 pada 29 jenis penyakit dengan akurasi hingga 99%, memberikan dasar teknis dalam menangani dataset dengan variasi kelas tinggi, tetapi belum menyertakan pembandingan arsitektur ringkas seperti MobileNetV2 [3]. Agustin, Bianto, dan Ardiansyah (2026) membuktikan CNN (akurasi 85%) mengungguli SVM (78%) dalam klasifikasi penyakit daun cabai [4], mengonfirmasi keunggulan ekstraksi fitur otomatis CNN dibandingkan fitur tangan pada skala dataset terbatas. Tinjauan literatur sistematis oleh Shafik dkk. (2023) terhadap ratusan studi deteksi penyakit tanaman mengungkapkan bahwa keterbatasan utama penelitian-penelitian sebelumnya terletak pada variasi ukuran dan keragaman dataset yang digunakan [5], sejalan dengan temuan Barbedo (2018) [6] bahwa efektivitas *transfer learning* sangat dipengaruhi oleh jumlah dan variasi sampel citra pelatihan yang tersedia untuk setiap kelas. Pada sisi arsitektur deteksi, Aldakheel dkk. (2024) [7] menerapkan YOLOv4 untuk deteksi dan identifikasi penyakit daun tanaman dengan hasil yang kompetitif, sedangkan Moyazzoma dkk. (2021) [8] secara khusus menerapkan MobileNetV2 melalui *transfer learning* berbasis ekstraksi fitur pra-latih untuk deteksi penyakit daun namun hanya pada skema klasifikasi biner (sehat/sakit), konsisten dengan tinjauan Li, Zhang, dan Wang (2021) yang menegaskan keunggulan pendekatan *deep learning* dibandingkan metode berbasis fitur tangan (*hand-crafted feature*) konvensional [9]. Pengembangan arsitektur yang ringkas juga menjadi perhatian pada penelitian Mazumder dkk. (2024) yang mengombinasikan DenseNet121 dan MobileNetV2 untuk deteksi penyakit lintas-spesies tanaman dengan data terbatas [10], serta Zhou dkk. (2024) yang memodifikasi ShuffleNetV2 agar lebih ringkas tanpa mengorbankan akurasi, keduanya relevan dengan kebutuhan *deployment* pada perangkat mobile di lapangan pertanian [11], namun keduanya belum menguji secara eksplisit interaksi antara ukuran batch dan strategi pelatihan. Di sisi lain, arsitektur berbasis Vision Transformer mulai dieksplorasi oleh Han dan Guo (2024) untuk klasifikasi penyakit daun tanaman berkayu (*ligneous*), menunjukkan potensi arsitektur non-konvolusional sebagai alternatif pengembangan di masa depan, meski dengan kebutuhan data dan komputasi yang jauh lebih besar [12]. Meskipun demikian, mayoritas penelitian tersebut masih berfokus pada satu atau beberapa komoditas tanaman dengan jumlah kelas penyakit yang relatif terbatas, serta jarang mengevaluasi secara eksplisit sensitivitas arsitektur terhadap hyperparameter pelatihan seperti ukuran batch (*batch size*). Terdapat pula *research gap* terkait penanganan ketidakseimbangan kelas pada dataset multikomoditas berskala besar, di mana pendekatan pembobotan kelas berbasis frekuensi seperti yang diusulkan Cui dkk. Pada tahun 2019, belum banyak diadopsi secara eksplisit pada studi identifikasi penyakit tanaman berskala besar [13]. Kebutuhan pengujian pada skala data yang jauh lebih besar ini sejalan dengan studi Ferentinos pada tahun 2018 yang menguji CNN pada lebih dari 87.000 citra dari 58 kelas tanaman dan mencapai akurasi 99,53% pada kondisi laboratorium [14], namun akurasi tersebut turun signifikan pada kondisi lapangan menegaskan pentingnya pengujian pada dataset yang merepresentasikan variasi dunia nyata.

Secara lebih spesifik, keterbatasan teknis pada studi-studi tersebut dapat dipetakan pada tiga aspek yang saling berkaitan. Pertama, dari sisi skala pengujian, mayoritas penelitian pembandingan arsitektur [2], [3], [7] hanya diuji pada satu hingga tiga puluh kelas penyakit dari satu komoditas, sehingga belum diketahui apakah urutan performa antararsitektur yang mereka laporkan tetap konsisten pada skala klasifikasi puluhan kali lebih besar dan lintas-komoditas seperti pada penelitian ini. Kedua, dari sisi hyperparameter, tidak satu pun studi di atas mengevaluasi secara sistematis interaksi antara strategi pelatihan (*from scratch* dibandingkan *Transfer Learning*) dengan konfigurasi *batch size*, padahal kerangka teoretis Keskar dkk. secara eksplisit memprediksi bahwa arah pengaruh *batch size* seharusnya bergantung pada titik awal optimisasi [15], suatu prediksi yang belum diuji secara empiris pada domain identifikasi penyakit tanaman. Ketiga, dari sisi validasi kualitatif, sebagian besar studi pembandingan arsitektur mengandalkan metrik akurasi agregat tanpa instrumen *Explainable AI* untuk memverifikasi apakah performa tinggi yang dicapai benar-benar merepresentasikan fitur visual penyakit yang bermakna secara agronomis, atau sekadar *overfitting* terhadap artefak dataset. Ketiga limitasi teknis inilah yang secara spesifik coba dijawab oleh penelitian ini melalui pengujian multikomoditas berskala besar, evaluasi sistematis interaksi arsitektur–*batch size*, serta triangulasi validasi kuantitatif dengan tiga instrumen *Explainable AI* (Confusion Matrix, t-SNE, dan Grad-CAM).

Landasan teoretis yang mendasari evaluasi sensitivitas *batch size* dalam penelitian ini adalah kerangka *sharp-flat minima* yang diajukan Keskar dkk. Tahun 2017, yang menunjukkan secara teoritis dan empiris bahwa *batch size* besar cenderung mengarahkan optimisasi menuju titik minima yang tajam (*sharp minima*) dengan kemampuan generalisasi yang lebih rendah [15], sedangkan *batch size* kecil menghasilkan estimasi gradien yang lebih ber-noise sehingga cenderung menemukan titik minima yang lebih landai (*flat minima*) dengan generalisasi yang lebih baik. Kerangka ini mendasari hipotesis penelitian bahwa arah pengaruh *batch size* terhadap performa model seharusnya bergantung pada titik awal proses optimisasi, inisialisasi acak pada Custom CNN dibandingkan titik representasi fitur pra-latih ImageNet pada MobileNetV2 dan InceptionV3, suatu hipotesis yang secara eksplisit diuji melalui desain eksperimen tiga arsitektur × tiga konfigurasi *batch size* pada penelitian ini.

Berdasarkan uraian tersebut, terdapat *research gap* mengenai perbandingan model *custom* dibandingkan model *pre-trained* berskala besar pada dataset multikomoditas, sekaligus evaluasi sensitivitas arsitektur terhadap konfigurasi *batch size* dan efektivitas strategi penanganan ketidakseimbangan kelas, yang masih jarang dibahas secara eksplisit dalam satu kerangka penelitian yang sama.



Penelitian ini bertujuan menganalisis dan membandingkan performa Custom CNN, MobileNetV2, dan InceptionV3 pada dataset multikomoditas berskala besar untuk menentukan model paling optimal dalam mendukung teknologi pertanian cerdas (*smart farming*), sekaligus mengevaluasi sensitivitas ketiga arsitektur tersebut terhadap variasi ukuran *batch* dan efektivitas strategi pembobotan kelas dalam mengatasi ketidakseimbangan sampel antar kelas penyakit. Melalui pendekatan *Explainable AI*, penelitian ini juga bertujuan memberikan pemahaman yang lebih mendalam mengenai bagaimana masing-masing arsitektur mempelajari representasi fitur visual penyakit daun, sehingga rekomendasi model yang dihasilkan tidak hanya didasarkan pada metrik akurasi semata, tetapi juga pada *interpretability* dan kelayakan implementasi di lapangan.

Kontribusi utama penelitian ini terhadap literatur identifikasi penyakit tanaman berbasis *deep learning* adalah sebagai berikut: (1) pengujian sistematis tiga arsitektur CNN pada skala 72 kelas penyakit dari 21 komoditas tanaman sekaligus, jauh melampaui skala pengujian mayoritas penelitian sejenis yang berfokus pada satu hingga beberapa komoditas; (2) bukti empiris pertama sejauh penelusuran literatur yang dilakukan mengenai pola sensitivitas *batch size* yang berlawanan arah antara arsitektur yang dilatih *from scratch* dan arsitektur *Transfer Learning*, didukung oleh uji signifikansi statistik dan analisis *generalization gap* berbasis kurva *loss*; dan (3) triangulasi validasi kuantitatif dan kualitatif melalui tiga instrumen *Explainable AI* (Confusion Matrix, t-SNE, dan Grad-CAM) untuk memastikan performa tinggi yang dicapai mencerminkan representasi fitur visual penyakit yang bermakna, bukan sekadar artefak dataset.

2. METODOLOGI PENELITIAN

Tahapan penelitian secara menyeluruh, mulai dari pengumpulan data hingga penyusunan kesimpulan, digambarkan pada diagram alur berikut (Gambar 1). Alur ini dirancang untuk memastikan bahwa proses eksperimen kedua kelompok pendekatan Custom CNN yang dilatih dari awal dan MobileNetV2/InceptionV3 yang memanfaatkan Transfer Learning berjalan pada kondisi pengujian yang setara, mulai dari sumber data, skema pembagian dataset, hingga metrik evaluasi akhir, sehingga perbandingan performa antararsitektur pada Bab 3 dapat diinterpretasikan secara adil tanpa bias akibat perbedaan perlakuan data atau lingkungan eksperimen. Percabangan pada diagram merepresentasikan dua jalur eksperimen yang berjalan paralel namun bermuara pada tahap evaluasi dan analisis *Explainable AI* yang identik, sehingga temuan pola sensitivitas *batch size* yang berlawanan arah antar kelompok model dapat ditelusuri murni dari perbedaan strategi pelatihan, bukan dari perbedaan protokol pengujian. Secara ringkas, alur tersebut mencakup tujuh tahap berurutan: pengumpulan dan pembersihan data, praproses citra, perancangan dan pelatihan ketiga arsitektur pada tiga konfigurasi *batch size*, penyimpanan bobot model terbaik, evaluasi komparatif pada data uji, analisis *Explainable AI*, hingga penyusunan kesimpulan dan rekomendasi dengan rincian tiap tahap dijabarkan pada subbab berikut.

Untuk menjamin bahwa perbedaan performa yang teramati antararsitektur maupun antar konfigurasi *batch size* benar-benar mencerminkan karakteristik model dan bukan artefak dari perbedaan perlakuan eksperimen, sejumlah kontrol diterapkan secara konsisten di seluruh skema pengujian. Pertama, seluruh sembilan skema (tiga arsitektur × tiga *batch size*) menggunakan pembagian data latih, validasi, dan uji yang identik sebagaimana disajikan pada Tabel 1, sehingga tidak ada model yang diuntungkan atau dirugikan oleh komposisi sampel yang berbeda. Kedua, seluruh eksperimen dijalankan pada perangkat keras dan perangkat lunak yang sama (GPU NVIDIA GeForce RTX 2060 6 GB VRAM, CUDA Toolkit 11.2, cuDNN v8), sehingga variasi performa akibat perbedaan lingkungan komputasi dapat dikesampingkan. Ketiga, pipeline praproses citra *resizing*, *normalisasi*, dan *augmentasi* diterapkan secara seragam pada seluruh skema menggunakan parameter *augmentasi* yang identik, sehingga perbedaan hasil tidak dapat diatribusikan pada perlakuan data yang tidak konsisten. Keempat, strategi penanganan ketidakseimbangan kelas melalui pembobotan kelas berbasis frekuensi diterapkan secara seragam pada seluruh arsitektur melalui parameter *class_weight* pada proses pelatihan, sebagaimana dijelaskan pada Subbab 2.5. Kontrol-kontrol ini bersama-sama memastikan bahwa perbandingan performa pada Bab 3 termasuk pola sensitivitas *batch size* yang berlawanan arah antara Custom CNN dan model Transfer Learning dapat diatribusikan secara valid pada perbedaan filosofi pelatihan (*from scratch* dibandingkan *transfer learning*), bukan pada variabel pengganggu (*confounding variable*) lain di luar arsitektur dan konfigurasi *batch size* itu sendiri. Kelima konfigurasi tersebut dijalankan secara berurutan pada sesi komputasi yang sama tanpa jeda perubahan *environment*, sehingga versi pustaka (TensorFlow, Keras, dan CUDA) tetap konsisten dari eksperimen pertama hingga terakhir.

Penelitian ini juga memiliki batasan ruang lingkup yang perlu ditegaskan agar interpretasi hasil tidak melampaui konteks pengujian yang dilakukan. Seluruh citra yang digunakan merupakan citra daun tunggal hasil pemotretan *close-up* dari repositori digital (Kaggle, Mendeley Data, dan PlantVillage), sehingga performa model pada kondisi lapangan nyata dengan variasi pencahayaan, sudut pengambilan, latar belakang kompleks, atau daun yang tumpang tindih dengan bagian tanaman lain belum tercakup dalam evaluasi ini dan memerlukan pengujian lanjutan pada dataset *in-the-wild*. Selain itu, pengukuran performa difokuskan pada aspek akurasi klasifikasi dan *interpretability* model melalui *Explainable AI*, sedangkan aspek operasional seperti *inference latency*, konsumsi daya, dan kestabilan model pada perangkat *edge* sesungguhnya berada di luar cakupan penelitian ini sebagaimana telah disinggung pada bagian keterbatasan penelitian. Cakupan 21 komoditas dan 72 kelas penyakit yang digunakan juga mewakili sebagian dari keragaman komoditas pertanian Indonesia, sehingga generalisasi temuan pada komoditas atau jenis penyakit di luar cakupan dataset ini perlu dilakukan secara hati-hati.



Gambar 1. Alur dan Tahapan Metodologi Penelitian

Secara umum, penelitian ini dilaksanakan melalui enam tahapan utama sebagaimana diilustrasikan pada Gambar 1, dengan rincian sebagai berikut:

- Pengumpulan dataset citra daun tanaman dari berbagai repositori sumber.
- Preprocessing citra, meliputi data cleaning, resizing, normalisasi, dan augmentasi.
- Perancangan tiga arsitektur model: Custom CNN, MobileNetV2, dan InceptionV3.
- Pelatihan model pada tiga konfigurasi batch size (8, 16, dan 32).
- Evaluasi performa dan analisis Explainable AI (Confusion Matrix, t-SNE, Grad-CAM).
- Analisis hasil dan penyusunan kesimpulan.

Rincian setiap tahapan dijabarkan pada subbab berikut.

2.1 Objek dan Ruang Lingkup Penelitian

Objek penelitian ini mencakup tiga arsitektur *deep learning* dengan filosofi pelatihan berbeda: Custom CNN yang dilatih dari awal (*from scratch*) tanpa bobot pra-latih, serta MobileNetV2 dan InceptionV3 yang memanfaatkan strategi *Transfer Learning* dua fase (*Feature Extraction* dan *Fine-Tuning*) dari bobot pra-latih ImageNet. Ketiga arsitektur tersebut diuji pada objek data berupa citra digital daun tanaman untuk identifikasi penyakit daun pada 72 kelas penyakit dari 21 komoditas tanaman pertanian dan hortikultura. Variabel kontrol utama yang dievaluasi adalah tiga konfigurasi *batch size* (8, 16, dan 32) guna mengungkap sensitivitas tiap arsitektur terhadap ukuran batch pelatihan. Untuk memvalidasi performa model secara kuantitatif sekaligus kualitatif, penelitian ini menerapkan tiga instrumen *Explainable AI* (Confusion Matrix, t-SNE, dan Grad-CAM) guna memverifikasi representasi fitur dan atensi visual yang dipelajari tiap model. Ruang lingkup penelitian dibatasi pada evaluasi berbasis citra statis (bukan video *real-time*) dan tidak mencakup pengukuran langsung waktu inferensi maupun konsumsi energi pada perangkat mobile/*edge*, sebagaimana dijelaskan pada bagian keterbatasan penelitian di Subbab 4.

2.2 Dataset dan Pengumpulan Data

Dataset penelitian ini merupakan data sekunder berupa citra digital daun tanaman dengan total 72.000 citra yang terbagi ke dalam 72 kelas penyakit dari 21 komoditas. Data dihimpun dari repositori internasional meliputi Kaggle (Potato Leaf Disease, Plant Pathology 2020, Rice Leaf Disease, Esca Grapes), Mendeley Data (tebu, kopi, tomat, terong, jeruk), dan



PlantVillage Dataset. Dataset dibagi dengan rasio mendekati 80:10:10 untuk data latih (57.682 citra), validasi (7.158 citra), dan uji (7.160 citra). Tahap prapemrosesan meliputi data cleaning dari 123.120 citra mentah, resizing ke 256×256 piksel, normalisasi nilai piksel, dan data augmentation (rotasi, zoom, horizontal flip, shear).

Proses *data cleaning* dilakukan pada tahap kurasi dataset, sebelum citra dibagi ke direktori data latih, validasi, dan uji, melalui tiga kriteria eksklusi: (1) verifikasi integritas berkas (penghapusan citra yang rusak, tidak dapat dibaca, atau berformat non-standar) pada seluruh 123.120 citra mentah hasil unduhan dari Kaggle, Mendeley Data, dan PlantVillage; (2) penghapusan citra beresolusi terlalu rendah untuk diproses secara bermakna pada ukuran input 256×256 piksel; dan (3) penyeragaman kuota tiap kelas penyakit menjadi 1.000 citra (800 latih : 100 validasi : 100 uji) melalui *random sampling* tanpa pengembalian pada kelas dengan jumlah citra berlebih, sebagaimana tercermin pada distribusi seragam 800:100:100 di Tabel 1. Kriteria ketiga inilah yang secara dominan menjelaskan pengurangan dari 123.120 menjadi 72.000 citra, mengingat variasi jumlah citra mentah yang cukup signifikan antar-repositori sumber data. Pembagian ke subset train/validation/test dilakukan pada tahap kurasi ini juga sebelum proses augmentasi sehingga tidak ada citra yang sama (atau turunan augmentasinya) muncul pada lebih dari satu subset.

Komposisi lengkap dataset per komoditas dan pembagiannya ke dalam data latih, validasi, dan uji disajikan pada Tabel 1. Setiap kelas penyakit memiliki proporsi 800:100:100 citra (80:10:10) secara seragam, kecuali kelas Padi Sehat yang memiliki proporsi 882:58:60 akibat karakteristik hasil proses cleaning pada repositori sumber data, sehingga total data latih, validasi, dan uji keseluruhan masing-masing berjumlah 57.682, 7.158, dan 7.160 citra.

Tabel 1. Komposisi Dataset per Komoditas (Data Latih, Validasi, dan Uji)

No	Kategori / Penyakit	Train (80%)	Val (10%)	Test (10%)	Total Citra
1	Anggur Busuk Hitam	800	100	100	1000
2	Anggur Campak Hitam	800	100	100	1000
3	Anggur Hawar Daun	800	100	100	1000
4	Anggur Sehat	800	100	100	1000
5	Apel Bercak Alternaria	800	100	100	1000
6	Apel Bercak Cokelat	800	100	100	1000
7	Apel Busuk Hitam	800	100	100	1000
8	Apel Kudis	800	100	100	1000
9	Apel Sehat	800	100	100	1000
10	Blueberry Sehat	800	100	100	1000
11	Ceri Embun Tepung	800	100	100	1000
12	Ceri Sehat	800	100	100	1000
13	Jagung Bercak Abu	800	100	100	1000
14	Jagung Hawar Utara	800	100	100	1000
15	Jagung Karat Daun	800	100	100	1000
16	Jagung Sehat	800	100	100	1000
17	Jambu Daun Kuning	800	100	100	1000
18	Jeruk Citrus Greening	800	100	100	1000
19	Jeruk Daun Rusak	800	100	100	1000
20	Jeruk Embun Tepung	800	100	100	1000
21	Jeruk Kanker	800	100	100	1000
22	Jeruk Kutu Kebul Berduri	800	100	100	1000
23	Jeruk Kutu Putih	800	100	100	1000
24	Jeruk Sehat	800	100	100	1000
25	Kedelai Sehat	800	100	100	1000
26	Kentang Hama	800	100	100	1000
27	Kentang Hawar Dini	800	100	100	1000
28	Kentang Hawar Lambat	800	100	100	1000
29	Kentang Layu Bakteri	800	100	100	1000
30	Kentang Nematoda	800	100	100	1000
31	Kentang Sehat	800	100	100	1000
32	Kentang Virus Daun Menggulung	800	100	100	1000
33	Kentang Virus Mosaik	800	100	100	1000
34	Kopi Bercak Phoma	800	100	100	1000
35	Labu Embun Tepung	800	100	100	1000
36	Mawar Karat Daun	800	100	100	1000
37	Padi Bercak Cokelat	800	100	100	1000
38	Padi Sehat	882	58	60	1000
39	Padi Tungro	800	100	100	1000
40	Paprika Bercak Bakteri	800	100	100	1000
41	Paprika Sehat	800	100	100	1000



No	Kategori / Penyakit	Train (80%)	Val (10%)	Test (10%)	Total Citra
42	Persik Bercak Bakteri	800	100	100	1000
43	Persik Sehat	800	100	100	1000
44	Raspberry Sehat	800	100	100	1000
45	Stroberi Daun Terbakar	800	100	100	1000
46	Stroberi Sehat	800	100	100	1000
47	Tebu Busuk Merah	800	100	100	1000
48	Tebu Daun Kuning	800	100	100	1000
49	Tebu Karat Daun	800	100	100	1000
50	Tebu Mosaik	800	100	100	1000
51	Tebu Sehat	800	100	100	1000
52	Terong Sehat	800	100	100	1000
53	Tomat Bercak Bakteri	800	100	100	1000
54	Tomat Bercak Septoria	800	100	100	1000
55	Tomat Bercak Target	800	100	100	1000
56	Tomat Daun Keriting	800	100	100	1000
57	Tomat Hawar Dini	800	100	100	1000
58	Tomat Hawar Lambat	800	100	100	1000
59	Tomat Jamur Daun	800	100	100	1000
60	Tomat Sehat	800	100	100	1000
61	Tomat Tungau Laba-laba	800	100	100	1000
62	Tomat Virus Mosaik	800	100	100	1000
63	jambu antraknosa	800	100	100	1000
64	jambu gigitan serangga	800	100	100	1000
65	jambu sehat	800	100	100	1000
66	jambu terbakar	800	100	100	1000
67	kopi karat daun	800	100	100	1000
68	kopi sehat	800	100	100	1000
69	mangga antraknosa	800	100	100	1000
70	mangga embun tepung	800	100	100	1000
71	mangga jelaga hitam	800	100	100	1000
72	mangga sehat	800	100	100	1000
Total					72.000

2.3 Arsitektur Model dan Strategi Pelatihan

Penelitian ini membandingkan tiga skenario arsitektur dengan strategi pelatihan yang berbeda, sebagaimana diilustrasikan pada Gambar 2, Gambar 3, dan Gambar 4.

2.3.1 Custom CNN

Custom CNN dibangun dari awal (*from scratch*) dengan lima blok konvolusi berurutan (32, 64, 128, 256, dan 512 filter). Setiap blok terdiri atas dua lapisan Conv2D berkernel 3×3 dengan regularisasi L2 (1×10^{-5}), diikuti Batch Normalization dan aktivasi ReLU pada masing-masing lapisan, kemudian MaxPooling2D (pool size 2) dan Dropout (0,10–0,30, meningkat setiap blok) untuk mencegah overfitting. Keluaran blok konvolusi terakhir (last_conv, dipakai sebagai basis Grad-CAM) diteruskan ke Global Average Pooling2D dan Batch Normalization, lalu ke Dense Layer 512 neuron beraktivasi ReLU (feature_dense) yang diikuti Batch Normalization dan Dropout 0,40. Lapisan keluaran berupa Dense 72 neuron beraktivasi Softmax, sesuai jumlah kelas penyakit. Model dilatih 80 epoch menggunakan optimizer Adam (learning rate 0,0002) dengan early stopping (patience=10).

2.3.2 MobileNetV2

MobileNetV2 (Sandler dkk., 2018) [16] menggunakan strategi Transfer Learning dua fase. Pada Fase 1 (Feature Extraction, 40 epoch, lr=0,0001), seluruh bobot backbone MobileNetV2 hasil pra-pelatihan ImageNet dibekukan (non-trainable), sedangkan lapisan yang ditambahkan di atasnya Global Average Pooling2D, Batch Normalization, Dense 512 neuron ReLU (feature_dense), Batch Normalization, Dropout 0,30, dan Dense 72 neuron Softmax dilatih dari awal. Pada Fase 2 (Fine-Tuning, 25 epoch, lr= 1×10^{-5}), 50 lapisan terakhir backbone (dari total 154 lapisan) dibuka kembali agar bobotnya dapat disesuaikan lebih halus terhadap karakteristik citra daun, sementara lapisan-lapisan awal tetap dibekukan untuk mempertahankan fitur generik hasil pra-pelatihan.

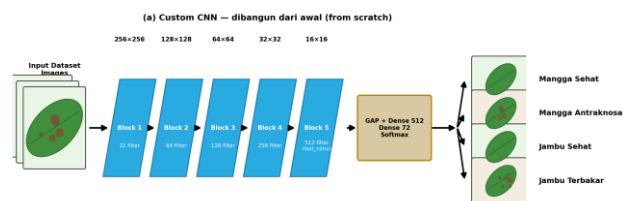
2.3.3 InceptionV3

InceptionV3 (Szegedy dkk., 2016) [17] menerapkan strategi serupa dengan MobileNetV2. Pada Fase 1, backbone InceptionV3 (hasil pra-pelatihan ImageNet) dibekukan sepenuhnya, dan hanya kepala klasifikasi (Global Average Pooling2D, Batch Normalization, Dense 512 neuron ReLU, Batch Normalization, Dropout 0,30, Dense 72 neuron

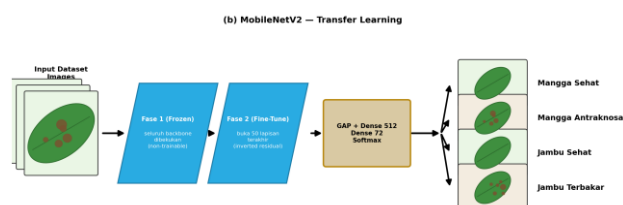
Softmax) yang dilatih. Pada Fase 2, 50 lapisan terakhir backbone (dari total 311 lapisan, dimulai dari blok mixed7 hingga mixed10) dibuka untuk fine-tuning. Seluruh model Custom CNN, MobileNetV2, dan InceptionV3 menggunakan mekanisme callbacks yang sama, meliputi EarlyStopping (patience=8--10), ReduceLROnPlateau (factor=0,2--0,3), dan ModelCheckpoint untuk menyimpan bobot terbaik, serta optimizer AdamW (Kingma dan Ba, 2015) [18] yang mengadaptasi laju pembelajaran tiap parameter secara individual. Ketidakeimbangan residual jumlah sampel antar kelas diatasi menggunakan strategi class weights, sebagaimana dijelaskan secara rinci pada Subbab 2.5.

Pemilihan 50 lapisan terakhir sebagai cakupan *fine-tuning* pada MobileNetV2 dan InceptionV3 bukan nilai arbitrer, melainkan hasil pertimbangan antara dua pendekatan ekstrem: membuka seluruh backbone berisiko *catastrophic forgetting* terhadap fitur generik hasil pra-pelatihan ImageNet dan menuntut *learning rate* yang sangat kecil untuk tetap stabil, sedangkan membekukan seluruh backbone membatasi kemampuan model beradaptasi terhadap karakteristik citra daun yang berbeda dari domain ImageNet. Lapisan 50 terakhir pada kedua arsitektur berada pada blok konvolusi tingkat tinggi (untuk InceptionV3, dimulai dari blok *mixed7* hingga *mixed10*) yang secara arsitektural merepresentasikan fitur semantik spesifik-domain, sementara lapisan-lapisan awal yang tetap dibekukan mempertahankan fitur generik tingkat rendah (tepi, tekstur, warna) yang cenderung dapat digunakan ulang lintas domain citra. Skema ini konsisten dengan praktik umum *Transfer Learning* dua fase, dan secara empiris pada eksperimen ini terbukti stabil pada seluruh konfigurasi *batch size* yang diuji (ditunjukkan tidak adanya kegagalan konvergensi pada Tabel 2), tanpa perlu pencarian hyperparameter tambahan berupa jumlah lapisan *fine-tuning* yang berbeda-beda; layer *Batch Normalization* tetap dibekukan meskipun berada dalam rentang 50 lapisan tersebut, untuk menjaga stabilitas statistik normalisasi saat inferensi. Pencegahan *data leakage* dijamin melalui dua mekanisme: pertama, pembagian citra ke direktori data latih, validasi, dan uji dilakukan pada tahap kurasi awal (Subbab 2.2), sebelum proses augmentasi dijalankan, sehingga tidak ada citra atau turunannya yang muncul pada lebih dari satu subset; kedua, augmentasi citra (*random flip*, *brightness*, *contrast*, *saturation*, *hue*, dan *random crop*) diterapkan secara *on-the-fly* hanya pada data latih melalui *pipeline tf.data*, sedangkan data validasi dan uji hanya melalui *resizing* dan normalisasi tanpa augmentasi apa pun, sehingga metrik evaluasi pada data uji murni mencerminkan generalisasi model terhadap citra yang belum pernah dilihat dalam bentuk apa pun selama pelatihan.

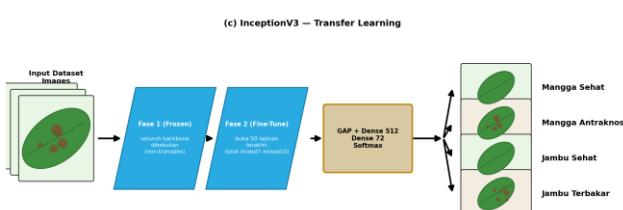
Berbeda dengan MobileNetV2 dan InceptionV3 yang bergantung pada bobot pra-latih, Custom CNN dirancang sepenuhnya dari nol dengan lima blok konvolusi berurutan (Gambar 2) yang secara progresif mengecilkan resolusi spasial citra dari 256×256 hingga 16×16 piksel sembari memperbanyak jumlah filter dari 32 pada blok pertama hingga 512 pada blok kelima, mengikuti pola umum arsitektur CNN di mana representasi tingkat rendah (tepi, tekstur) dipelajari pada lapisan awal dan representasi tingkat tinggi (bentuk lesi, pola penyakit) dipelajari pada lapisan akhir. Total parameter Custom CNN yang dihasilkan dari desain ini adalah $\pm 5,02$ juta, lebih besar dibandingkan MobileNetV2 ($\pm 2,96$ juta) namun jauh lebih ringkas dibandingkan InceptionV3 ($\pm 22,90$ juta), sebagaimana dirangkum pada Tabel 3. Setiap blok konvolusi dilengkapi Batch Normalization untuk menstabilkan distribusi aktivasi antar-lapisan serta Dropout untuk mengurangi risiko overfitting, mengingat model ini tidak memiliki keuntungan inisialisasi bobot yang sudah mengandung representasi visual umum sebagaimana dimiliki kedua model Transfer Learning.



Gambar 2. Arsitektur Custom CNN yang Digunakan dalam Penelitian



Gambar 3. Arsitektur MobileNetV2 yang Digunakan dalam Penelitian



Gambar 4. Arsitektur InceptionV3 yang Digunakan dalam Penelitian



2.4 Metrik Evaluasi

Evaluasi menggunakan metrik kuantitatif: Accuracy (persentase prediksi benar), Precision (rasio prediksi positif tepat), Recall (kemampuan menemukan sampel positif), dan F1-Score (rata-rata harmonik Precision dan Recall), sebagaimana dirumuskan pada Persamaan (1)–(4), serta Loss. Instrumen Explainable AI (XAI) meliputi Confusion Matrix untuk analisis kesalahan klasifikasi, t-SNE (Maaten dan Hinton, 2008) [19] untuk visualisasi pemisahan fitur antar kelas, dan Grad-CAM (Selvaraju dkk., 2017) [20] untuk analisis area atensi model pada citra daun. Seluruh eksperimen dijalankan pada GPU NVIDIA GeForce RTX 2060 (6 GB VRAM) dengan CUDA Toolkit 11.2 dan cuDNN v8.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

$$Precision = TP / (TP + FP) \quad (2)$$

$$Recall = TP / (TP + FN) \quad (3)$$

$$F1-Score = 2 \times (Precision \times Recall) / (Precision + Recall) \quad (4)$$

2.5 Penanganan Ketidakseimbangan Kelas (Class Imbalance)

Meskipun proses kurasi menghasilkan dataset akhir yang relatif merata 72.000 citra terbagi ke dalam 72 kelas penyakit (rata-rata ± 1.000 citra per kelas) dari total 123.120 citra mentah yang telah melalui tahap data cleaning variasi residual jumlah sampel antar kelas tetap muncul akibat perbedaan ketersediaan citra pada masing-masing repositori sumber (Kaggle, Mendeley Data, dan PlantVillage). Berdasarkan komposisi dataset pada Tabel 1, variasi ini secara spesifik hanya bersumber dari satu kelas, yaitu Padi Sehat, yang memiliki proporsi pembagian 882:58:60 (data latih:validasi:uji), berbeda dari proporsi seragam 800:100:100 yang berlaku pada 71 kelas lainnya. Tanpa penanganan, variasi ini berpotensi membuat model bias terhadap kelas dengan jumlah sampel lebih banyak, terutama pada kelas-kelas dengan kemiripan visual tinggi antarpenyakit.

Strategi yang digunakan untuk mengatasi ketidakseimbangan tersebut adalah pembobotan kelas (class weight) berbasis frekuensi terbalik (inverse-frequency), dihitung menggunakan fungsi `compute_class_weight` (mode `'balanced'`) dari pustaka `scikit-learn` terhadap data latih (57.600 citra). Bobot tiap kelas c dirumuskan sebagai $w(c) = n_{\text{total}} / (K \times n(c))$, dengan n_{total} adalah jumlah total sampel data latih, K adalah jumlah kelas (72), dan $n(c)$ adalah jumlah sampel pada kelas c . Untuk mencegah kelas dengan sampel sangat sedikit memperoleh bobot berlebihan sehingga mendominasi gradien saat pelatihan, bobot dibatasi (clipping) pada nilai maksimal 5,0. Skema pembobotan ini diterapkan secara identik pada seluruh arsitektur Custom CNN, MobileNetV2, dan InceptionV3 melalui parameter `class_weight` pada proses `model.fit()`, sehingga perbandingan performa antararsitektur tidak dipengaruhi oleh perbedaan perlakuan terhadap ketidakseimbangan kelas.

Hasil perhitungan menunjukkan bobot kelas pada data latih berkisar antara 0,9083 hingga 1,0014 (rata-rata 1,0001) untuk seluruh 72 kelas, mengindikasikan bahwa dataset hasil kurasi sudah mendekati seimbang sehingga ambang clipping 5,0 pada praktiknya tidak pernah tercapai. Nilai bobot terendah (0,9083) tercatat pada kelas Padi Sehat, konsisten dengan temuan pada Tabel 1 bahwa kelas tersebut justru memiliki jumlah data latih paling banyak (882 citra) dibandingkan 71 kelas lainnya yang seragam pada 800 citra, sehingga sesuai prinsip pembobotan berbasis frekuensi terbalik, kelas ini memperoleh bobot sedikit di bawah 1,0 untuk mengimbangi kontribusinya yang secara alami sudah lebih dominan terhadap fungsi loss; sebaliknya, ke-71 kelas dengan 800 citra memperoleh bobot seragam tertinggi (1,0014). Dengan demikian, pembobotan kelas berfungsi lebih sebagai mekanisme pencegahan (safety mechanism) terhadap bias residual dibandingkan sebagai koreksi berat terhadap ketidakseimbangan ekstrem. Hal ini turut menjelaskan mengapa nilai F1-Score (macro) pada Tabel 2 secara konsisten mendekati nilai test accuracy di seluruh skema eksperimen, karena performa kelas minoritas tidak tenggelam oleh dominasi kelas mayoritas dalam perhitungan metrik rata-rata makro.

Meskipun demikian, ketidakseimbangan kelas berpotensi menjadi lebih signifikan pada perluasan penelitian ke komoditas atau sumber data lain dengan jumlah citra yang lebih timpang. Oleh karena itu, sebagaimana disebutkan pada bagian kesimpulan, penggunaan teknik augmentasi sintesis berbasis Generative Adversarial Network (GAN) untuk kelas minoritas menjadi arah pengembangan lanjutan yang relevan untuk memperkuat ketahanan model terhadap ketidakseimbangan kelas yang lebih ekstrem.

3. HASIL DAN PEMBAHASAN

Sebagai pengingat singkat, seluruh eksperimen pada bagian ini menggunakan objek dan komposisi dataset yang telah dijelaskan pada Subbab 2.1–2.2 (72.000 citra, 72 kelas penyakit dari 21 komoditas, dengan rincian pembagian train/validation/test per komoditas pada Tabel 1). Eksperimen perbandingan ketiga arsitektur Custom CNN, MobileNetV2, dan InceptionV3 dilaksanakan pada lingkungan yang sama menggunakan GPU NVIDIA GeForce RTX 2060 (6 GB VRAM), dengan ukuran citra input seragam 256×256 piksel. Untuk mengevaluasi sensitivitas tiap arsitektur terhadap ukuran batch, setiap model dilatih dan diuji pada tiga konfigurasi batch size, yaitu 8, 16, dan 32, sehingga menghasilkan sembilan skema eksperimen. Evaluasi akhir dilakukan pada data uji (test set) sebanyak 7.160 citra menggunakan bobot model terbaik (best checkpoint) hasil early stopping pada tiap skema. Skala pengujian pada penelitian ini, 72 kelas penyakit dari 21 komoditas tergolong jauh lebih luas dibandingkan sebagian besar penelitian sejenis yang umumnya berfokus hanya pada satu hingga beberapa komoditas tanaman saja [2], [3], [4], sehingga hasil perbandingan

performa pada penelitian ini diharapkan dapat merepresentasikan kondisi multikomoditas yang lebih mendekati skenario penerapan nyata di lapangan pertanian Indonesia.

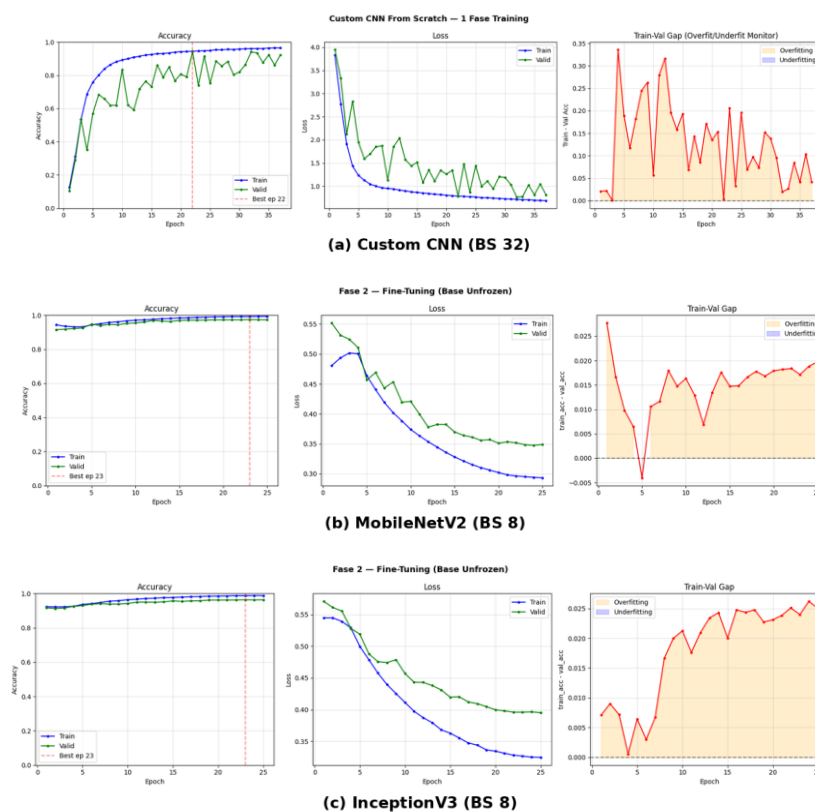
3.1 Perbandingan Hasil Pelatihan Model Berdasarkan Batch Size

Tabel 2 merekap validation accuracy, validation loss, test accuracy, dan F1-Score (macro) ketiga arsitektur pada tiap konfigurasi batch size. Custom CNN yang dilatih from scratch menunjukkan pola peningkatan performa yang konsisten seiring pertambahan batch size, dari test accuracy 89,61% (batch size 8) menjadi 92,70% (batch size 16), hingga mencapai puncaknya 94,72% (batch size 32). Pola sebaliknya justru teramati pada kedua model Transfer Learning: MobileNetV2 mencapai performa terbaik pada batch size 8 (test accuracy 97,57%, F1-Score 0,9755) dan menurun bertahap pada batch size 16 (97,37%) serta batch size 32 (96,77%). InceptionV3 menunjukkan kecenderungan serupa, dengan test accuracy tertinggi 96,35% pada batch size 8, sedikit menurun menjadi 96,30% (batch size 16) dan 95,99% (batch size 32). Capaian test accuracy MobileNetV2 sebesar 97,57% pada penelitian ini turut sejalan dengan tren peningkatan akurasi pada studi-studi deteksi penyakit tanaman berbasis MobileNetV2 lainnya, seperti pada penelitian Moyazzoma dkk. (2021) [8], meskipun penelitian ini diuji pada jumlah kelas penyakit yang jauh lebih banyak (72 kelas) dibandingkan skala pengujian pada penelitian tersebut, menunjukkan konsistensi keunggulan arsitektur MobileNetV2 bahkan pada skenario klasifikasi multikelas berskala besar.

Tabel 2. Perbandingan Performa Ketiga Arsitektur pada Variasi Batch Size 8, 16, dan 32

Arsitektur	Batch Size	Val. Accuracy	Val. Loss	Test Accuracy	F1-Score (Macro)
Custom CNN	8	89,48%	1,0923	89,61%	0,8941
Custom CNN	16	92,11%	0,9183	92,70%	0,9266
Custom CNN	32	94,16%	0,7846	94,72%	0,9474
MobileNetV2	8	97,50%	0,3485	97,57%	0,9755
MobileNetV2	16	97,14%	0,3496	97,37%	0,9736
MobileNetV2	32	96,63%	0,3518	96,77%	0,9675
InceptionV3	8	96,34%	0,3961	96,35%	0,9635
InceptionV3	16	95,78%	0,3937	96,30%	0,9629
InceptionV3	32	95,64%	0,3876	95,99%	0,9599

Untuk melihat bagaimana proses pelatihan berlangsung pada konfigurasi terbaik masing-masing arsitektur, kurva akurasi dan loss selama pelatihan ditampilkan pada Gambar 5 berikut.



Gambar 5. Kurva Akurasi dan Loss Pelatihan pada Konfigurasi Batch Size Terbaik: (a) Custom CNN (BS 32), (b) MobileNetV2 (BS 8), (c) InceptionV3 (BS 8)

Berdasarkan Tabel 2, konfigurasi optimal berbeda-beda pada tiap arsitektur: Custom CNN mencapai hasil terbaik pada batch size 32, sedangkan MobileNetV2 dan InceptionV3 justru optimal pada batch size 8. Tabel 3 merekap perbandingan performa terbaik ketiga arsitektur menggunakan konfigurasi batch size optimalnya masing-masing.

3.2 Signifikansi Statistik Perbedaan Performa.

Untuk membuktikan bahwa perbedaan performa antar konfigurasi *batch size* dan antararsitektur bukan sekadar variasi acak, dilakukan uji proporsi dua sampel (*two-proportion z-test*) terhadap *test accuracy* pada data uji ($n=7.160$), yang sesuai untuk membandingkan proporsi prediksi benar antar dua kondisi pengujian independen. Pada Custom CNN, selisih akurasi antara batch size 8 (89,61%) dan batch size 32 (94,72%) terbukti signifikan secara statistik ($z=-11,38$, $p<0,0001$), demikian pula pada MobileNetV2 antara batch size 32 (96,77%) dan batch size 8 (97,57%) ($z=-2,87$, $p=0,0041$). Pada InceptionV3, selisih antara batch size 32 (95,99%) dan batch size 8 (96,35%) relatif kecil dan tidak signifikan secara statistik ($z=-1,13$, $p=0,257$), menunjukkan bahwa sensitivitas InceptionV3 terhadap *batch size* lebih lemah dibandingkan kedua arsitektur lainnya meskipun arah kecenderungannya tetap konsisten. Pada perbandingan antararsitektur di konfigurasi optimalnya masing-masing, MobileNetV2 (97,57%) unggul signifikan dibandingkan InceptionV3 (96,35%) ($z=4,24$, $p<0,0001$) maupun dibandingkan Custom CNN (94,72%) ($z=8,86$, $p<0,0001$). Hasil uji statistik ini mengonfirmasi bahwa pola sensitivitas *batch size* yang berlawanan arah antara Custom CNN dan kedua model *Transfer Learning* bukan sekadar fluktuasi pengujian tunggal, melainkan perbedaan performa yang secara matematis signifikan pada tingkat kepercayaan 95%. Perlu dicatat sebagai keterbatasan metodologis bahwa uji ini membandingkan proporsi akurasi agregat dan mengasumsikan independensi antar sampel uji, sehingga tidak menggantikan uji berpasangan (*McNemar's test*) pada tingkat prediksi per-sampel yang idealnya memerlukan data prediksi mentah (label prediksi tiap citra uji) dari kesembilan skema eksperimen; pengujian berpasangan tersebut disarankan sebagai penguatan pada penelitian lanjutan apabila data prediksi per-sampel tersedia.

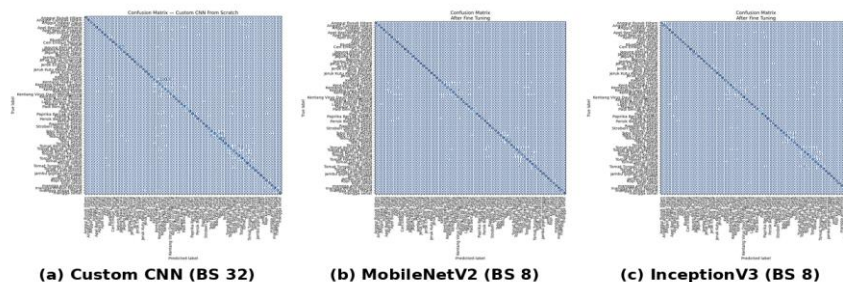
Tabel 3. Rekap Perbandingan Performa Terbaik Ketiga Arsitektur

Metrik	Custom CNN (BS 32)	MobileNetV2 (BS 8)	InceptionV3 (BS 8)
Val. Accuracy	94,16%	97,50%	96,34%
Val. Loss	0,7846	0,3485	0,3961
Test Accuracy	94,72%	97,57%	96,35%
F1-Score (Macro)	0,9474	0,9755	0,9635
Total Parameter	$\pm 5,02$ juta	$\pm 2,96$ juta	$\pm 22,90$ juta

3.3 Analisis Explainable AI (XAI)

Analisis classification report pada konfigurasi terbaik MobileNetV2 (batch size 8) menunjukkan tidak ada lagi kelas dengan performa sangat rendah seperti pada eksperimen sebelumnya; kelas tersulit yang teridentifikasi seperti Tomat Hawar Dini (F1-Score 0,8934) dan Kentang Sehat (F1-Score 0,8995) tetap mencapai F1-Score di atas 0,89, sementara 20 dari 72 kelas mencapai F1-Score sempurna (1,00), antara lain kelas Jambu Sehat, Jambu Terbakar, Mangga Antraknosa, dan Mangga Sehat. Pola ini mengindikasikan representasi fitur MobileNetV2 pada batch size kecil mampu memisahkan kelas-kelas dengan kemiripan visual tinggi secara lebih baik dibandingkan pada batch size besar. Kelas-kelas yang masih menunjukkan F1-Score relatif lebih rendah umumnya merupakan kelas dengan gejala awal penyakit (*early-stage symptoms*) yang secara visual belum terlalu berbeda dari kondisi daun sehat, sehingga tantangan klasifikasinya lebih bersifat inheren pada karakteristik citra itu sendiri dibandingkan keterbatasan arsitektur model. Secara umum, kedua model *Transfer Learning* (MobileNetV2 dan InceptionV3) tetap konsisten unggul dibandingkan Custom CNN pada seluruh konfigurasi batch size yang diuji, mengonfirmasi keunggulan strategi Feature Extraction dan Fine-Tuning dalam mengekstraksi fitur diskriminatif dari citra daun.

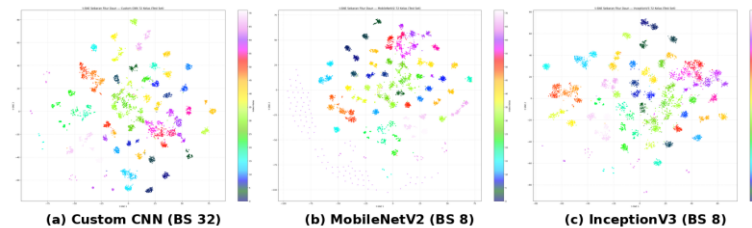
Untuk memverifikasi temuan kuantitatif tersebut secara visual, dilakukan analisis Explainable AI menggunakan Confusion Matrix (Gambar 6) dan visualisasi t-SNE (Gambar 7).



Gambar 6. Confusion Matrix pada Data Uji: (a) Custom CNN (BS 32), (b) MobileNetV2 (BS 8), (c) InceptionV3 (BS 8)

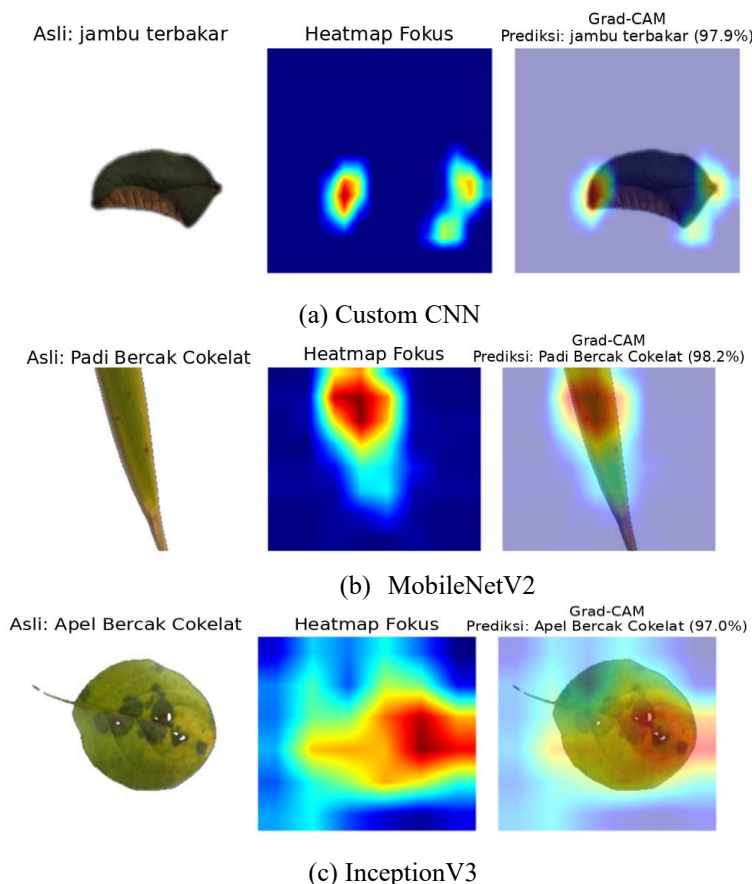
Confusion Matrix pada Gambar 6 memperlihatkan pola kesalahan klasifikasi yang berbeda antarmodel. Pada Custom CNN (Gambar 6a), kesalahan klasifikasi tersebar lebih merata di luar diagonal utama dan cenderung

terkonsentrasi pada pasangan kelas dengan kemiripan visual tinggi, misalnya antarvarian penyakit bercak daun pada komoditas yang sama, mengindikasikan bahwa model yang dilatih dari awal masih kesulitan membedakan tekstur lesi yang serupa tanpa bantuan representasi fitur pra-latih. Sebaliknya, MobileNetV2 (Gambar 6b) menunjukkan diagonal matriks yang jauh lebih tajam dengan kesalahan yang jarang dan tidak terpolat pada kelompok kelas tertentu, mengonfirmasi bahwa fitur hasil Transfer Learning mampu memisahkan kelas-kelas dengan gejala visual mirip secara lebih andal. InceptionV3 (Gambar 6c) berada pada posisi menengah diagonalnya cukup tajam namun terdapat beberapa sel kesalahan yang lebih menonjol dibandingkan MobileNetV2, konsisten dengan F1-Score makronya yang sedikit lebih rendah pada Tabel 3. Secara keseluruhan, pola pada Confusion Matrix ini memperkuat temuan kuantitatif bahwa MobileNetV2 tidak hanya unggul pada metrik agregat, tetapi juga pada tingkat konsistensi antar-kelas penyakit secara individual.



Gambar 7. Visualisasi t-SNE Fitur Antar Kelas: (a) Custom CNN (BS 32), (b) MobileNetV2 (BS 8), (c) InceptionV3 (BS 8)

Sebagai instrumen ketiga, area atensi model divisualisasikan menggunakan Grad-CAM pada Gambar 8 untuk mengonfirmasi apakah model benar-benar berfokus pada region lesi penyakit.



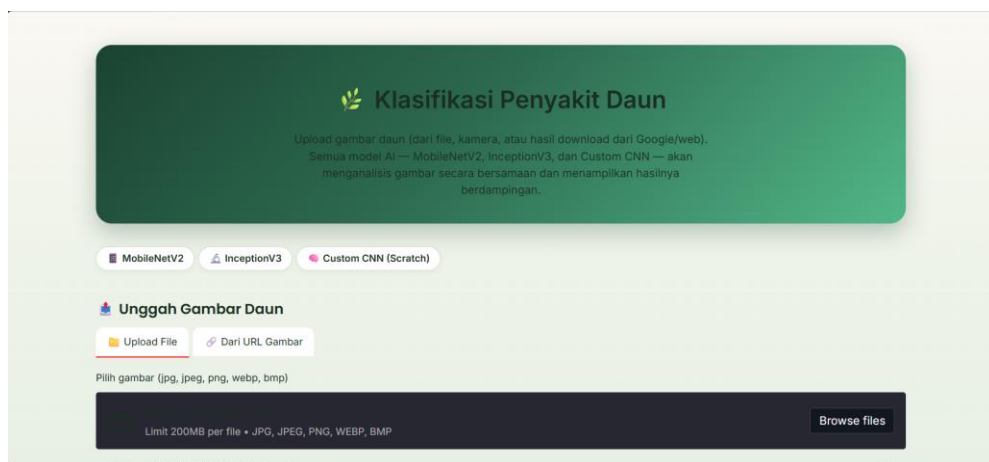
Gambar 8. Visualisasi Grad-CAM Area Atensi Model: (a) Custom CNN (BS 32), (b) MobileNetV2 (BS 8), (c) InceptionV3 (BS 8)

Konsistensi antara ketiga instrumen Explainable AI yang digunakan turut memperkuat validitas temuan ini. Pada Confusion Matrix (Gambar 6), MobileNetV2 menunjukkan diagonal matriks yang lebih tajam dengan kesalahan klasifikasi yang lebih tersebar merata dibandingkan Custom CNN, mengindikasikan tidak adanya bias sistematis terhadap kelas tertentu. Visualisasi t-SNE (Gambar 7) memperlihatkan bahwa fitur yang diekstraksi MobileNetV2 pada lapisan feature_dense membentuk kluster yang lebih rapat dan terpisah antar kelas dibandingkan Custom CNN, yang klusternya tampak lebih tumpang tindih terutama pada kelas-kelas dengan gejala visual serupa seperti berbagai jenis bercak daun

(*leaf spot*) pada komoditas berbeda. Pola ini konsisten dengan visualisasi Grad-CAM (Gambar 8) yang menunjukkan MobileNetV2 secara konsisten memusatkan atensi pada region lesi penyakit yang relevan secara agronomis, alih-alih pada area latar belakang atau tepi daun yang tidak informatif. Temuan ini memberikan keyakinan bahwa performa tinggi yang dicapai MobileNetV2 bukan disebabkan oleh overfitting terhadap artefak dataset, melainkan benar-benar merepresentasikan pembelajaran fitur visual penyakit yang bermakna secara agronomis sebuah aspek krusial bagi kelayakan model untuk diadopsi dalam sistem pendukung keputusan diagnosis penyakit tanaman di lapangan.

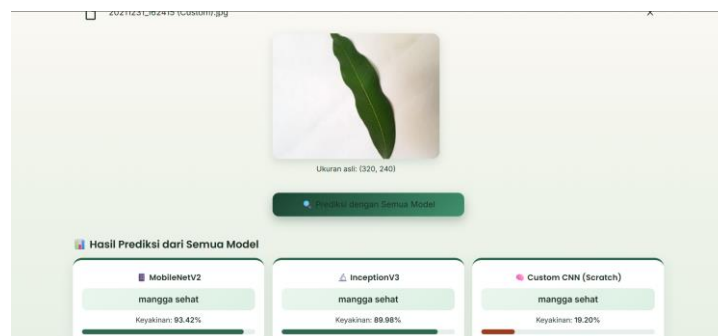
3.4 Implementasi dan Pengujian Deployment Model

Untuk mengevaluasi kelayakan penerapan praktis, ketiga model klasifikasi (MobileNetV2, InceptionV3, dan Custom CNN) diimplementasikan ke dalam sebuah aplikasi web berbasis Streamlit bernama "Klasifikasi Penyakit Daun". Aplikasi ini memungkinkan pengguna mengunggah citra daun baik melalui berkas lokal, kamera, maupun tautan URL gambar dalam format JPG, JPEG, PNG, WEBP, atau BMP, kemudian menjalankan inferensi ketiga model secara bersamaan dan menampilkan hasil prediksi berdampingan (*side-by-side*) sebagaimana ditunjukkan pada Gambar 9. Skema deployment multi-model ini dirancang untuk memberikan gambaran perbandingan performa antar arsitektur secara langsung pada skenario penggunaan nyata, sekaligus menguji konsistensi hasil ketiga model terhadap citra input di luar dataset pelatihan dan pengujian (*out-of-sample*).



Gambar 9. Antarmuka Aplikasi Web Deployment untuk Klasifikasi Penyakit Daun Berbasis Streamlit

Sebagai ilustrasi, Gambar 10 menyajikan salah satu kasus uji menggunakan citra daun jambu yang menunjukkan gejala bercak pada permukaannya. MobileNetV2 dan InceptionV3 secara konsisten memprediksi kelas "jambu antraknosa" dengan tingkat keyakinan masing-masing 63,31% dan 73,66%, sedangkan Custom CNN justru memprediksi kelas yang berbeda jauh secara komoditas, yaitu "Kentang Hama", dengan tingkat keyakinan yang jauh lebih rendah (26,94%). Kesepakatan (*agreement*) antara kedua model Transfer Learning pada kasus uji ini memperkuat temuan kuantitatif pada subbab sebelumnya bahwa MobileNetV2 dan InceptionV3 memiliki representasi fitur yang lebih matang dan konsisten dibandingkan Custom CNN. Sebaliknya, kesalahan klasifikasi lintas-komoditas serta rendahnya tingkat keyakinan yang ditunjukkan Custom CNN pada kasus ini mengindikasikan keterbatasan generalisasi model yang dilatih dari awal (*from scratch*) ketika berhadapan dengan citra dunia nyata (*in-the-wild*) yang kondisi pencahayaan, sudut pengambilan, dan latar belakangnya berbeda dari citra pada dataset pelatihan.



Gambar 10. Hasil Prediksi Ketiga Model pada Salah Satu Kasus Uji Deployment: MobileNetV2 dan InceptionV3 Memprediksi "Jambu Antraknosa", Sedangkan Custom CNN Memprediksi "Kentang Hama"

Hasil pengujian deployment ini memberikan bukti pendukung tambahan, di luar metrik kuantitatif pada data uji, bahwa MobileNetV2 layak dipertimbangkan sebagai kandidat utama untuk implementasi pada aplikasi mobile petani maupun sistem IoT pertanian, mengingat kombinasi antara akurasi prediksi yang tinggi, tingkat keyakinan yang lebih stabil, serta ukuran model yang jauh lebih ringkas dibandingkan InceptionV3. Meskipun demikian, pengujian deployment



pada penelitian ini masih bersifat kualitatif dan berbasis studi kasus tunggal, sehingga evaluasi yang lebih sistematis terhadap sejumlah besar citra in-the-wild, termasuk pengukuran waktu inferensi dan konsumsi sumber daya pada perangkat mobile, tetap diperlukan pada penelitian lanjutan sebagaimana disebutkan pada Bab Kesimpulan.

3.5 Implikasi Hasil

Temuan paling signifikan dari eksperimen variasi batch size adalah adanya pola sensitivitas yang berlawanan arah antara model yang dilatih from scratch dan model Transfer Learning. Custom CNN menunjukkan tren peningkatan performa seiring pertambahan batch size, kemungkinan karena batch yang lebih besar memberikan estimasi gradien yang lebih stabil pada proses pembelajaran fitur dari awal (*random initialization*) sehingga optimisasi berjalan lebih efektif. Sebaliknya, MobileNetV2 dan InceptionV3 yang telah memiliki representasi fitur awal dari bobot pre-trained ImageNet, justru lebih diuntungkan oleh batch size kecil (8), sejalan dengan temuan pada literatur bahwa batch kecil menghasilkan noise gradien yang membantu proses fine-tuning keluar dari local minima serta meningkatkan generalisasi model pre-trained.

Selain itu, keunggulan MobileNetV2 dibandingkan InceptionV3 tetap konsisten pada seluruh konfigurasi batch size, dengan margin test accuracy antara 0,4--1,2 poin persentase, meskipun jumlah parameter MobileNetV2 ($\pm 2,96$ juta) jauh lebih sedikit dibandingkan InceptionV3 ($\pm 22,90$ juta, atau hampir 8 kali lebih besar). Hal ini kembali menegaskan bahwa arsitektur yang lebih ringkas tidak selalu berkorelasi dengan performa yang lebih rendah, bahkan berpotensi lebih unggul ketika dikombinasikan dengan strategi batch size yang tepat. Selisih validation loss antara MobileNetV2 (0,3485) dan Custom CNN (0,7846) pada konfigurasi terbaik masing-masing lebih dari dua kali lipat juga mengindikasikan model Transfer Learning jauh lebih terkalibrasi dalam mengekspresikan confidence prediksinya, sebuah aspek yang krusial dalam sistem pendukung keputusan diagnosis penyakit tanaman.

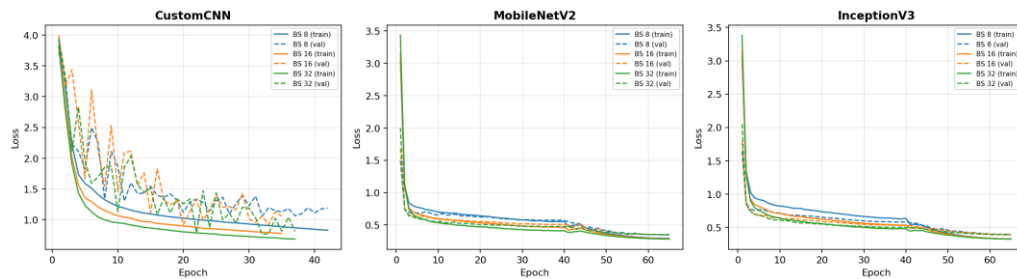
Pola sensitivitas batch size yang berlawanan arah antara Custom CNN dan kedua model Transfer Learning ini sejalan dengan kerangka teoretis yang diajukan Keskar dkk. (2017) [15], yang menunjukkan bahwa pelatihan dengan batch size besar cenderung konvergen pada titik minima yang tajam (*sharp minima*) sehingga rentan terhadap penurunan kemampuan generalisasi, sedangkan batch size kecil menghasilkan noise stokastik pada estimasi gradien yang mendorong optimisasi menuju titik minima yang lebih landai (*flat minima*) dan cenderung menggeneralisasi lebih baik pada data yang belum pernah dilihat model. Pada kasus Custom CNN yang dilatih dari awal, kondisi random initialization membuat lanskap loss pada tahap awal pelatihan jauh lebih kasar dan tidak stabil, sehingga estimasi gradien yang lebih presisi dari batch besar justru lebih diperlukan untuk mengarahkan optimisasi secara konsisten. Sebaliknya, MobileNetV2 dan InceptionV3 memulai pelatihan dari titik representasi fitur yang sudah relatif optimal hasil pra-pelatihan ImageNet, sehingga noise gradien dari batch kecil justru berperan sebagai bentuk regularisasi implisit yang membantu proses fine-tuning keluar dari local minima yang terlalu spesifik terhadap domain ImageNet dan bergerak menuju representasi yang lebih sesuai dengan karakteristik citra daun.

Temuan-temuan di atas secara langsung menjawab ketiga kontribusi utama yang diajukan pada Subbab 1: pengujian sistematis sensitivitas batch size pada skala 72 kelas penyakit dari 21 komoditas (kontribusi pertama) membuktikan bahwa arah pengaruh batch size bergantung pada filosofi pelatihan yang digunakan, bukan sekadar karakteristik arsitektur secara umum; validasi statistik melalui uji proporsi dua sampel dan analisis kurva loss (kontribusi kedua) memastikan bahwa pola berlawanan arah tersebut bukan artefak pengujian tunggal, melainkan perbedaan yang secara matematis dan mekanistik dapat dipertanggungjawabkan; serta triangulasi melalui Confusion Matrix, t-SNE, dan Grad-CAM (kontribusi ketiga) menunjukkan bahwa performa kuantitatif MobileNetV2 yang unggul juga tercermin secara kualitatif pada representasi fitur dan area atensi model, bukan sekadar hasil kebetulan pada satu metrik evaluasi. Implikasi praktisnya, praktisi yang hendak mengembangkan sistem identifikasi penyakit tanaman berbasis Transfer Learning disarankan memprioritaskan batch size kecil (8) untuk memaksimalkan performa, sementara pengembang yang memilih melatih arsitektur dari awal perlu mempertimbangkan batch size yang lebih besar (32) sembari tetap memantau kebutuhan memori GPU yang meningkat seiring pembesaran batch.

3.6 Analisis Kurva Loss dan Generalization Gap.

Untuk menguji klaim di atas secara empiris, bukan sekadar argumen teoretis, dilakukan analisis terhadap kurva *training loss* dan *validation loss* tiap epoch pada seluruh sembilan skema eksperimen (Gambar 11). Indikator yang digunakan adalah *generalization gap* (selisih *validation loss* dikurangi *training loss* pada epoch terbaik berdasarkan *val accuracy*), yang secara konseptual berkaitan dengan ketajaman titik minima yang dicapai: gap yang lebih besar mengindikasikan model overfit relatif terhadap data latih, konsisten dengan karakteristik *sharp minima*. Pada Custom CNN, gap menyempit secara konsisten seiring pembesaran *batch size*: 0,1385 (BS 8) $\rightarrow$ 0,0208 (BS 16) $\rightarrow$ -0,0001 (BS 32), menunjukkan generalisasi yang membaik monoton pada batch besar. Pola ini berbalik arah pada MobileNetV2, di mana gap justru melebar seiring pembesaran *batch size*: 0,0531 (BS 8) $\rightarrow$ 0,0580 (BS 16) $\rightarrow$ 0,0587 (BS 32). Pada InceptionV3, gap relatif stabil pada kisaran sempit (0,0635–0,0692) tanpa tren monoton yang jelas, konsisten dengan hasil uji signifikansi statistik pada Subbab 3.2 yang menunjukkan sensitivitas *batch size* arsitektur ini secara statistik tidak signifikan. Tren *generalization gap* pada ketiga arsitektur ini secara kuantitatif memperkuat, bukan sekadar mengulang, klaim pola berlawanan arah yang dilaporkan berdasarkan metrik akurasi, sekaligus memberi bukti tidak langsung yang konsisten dengan prediksi kerangka *sharp-flat minima* Keskar dkk. [15]. Sebagai catatan metodologis, analisis ini merupakan proksi berbasis kurva loss dan belum setara dengan pengukuran ketajaman lanskap loss secara langsung (misalnya melalui

estimasi nilai eigen Hessian atau visualisasi *loss landscape*), yang memerlukan akses langsung terhadap bobot model tersimpan dan disarankan sebagai penguatan pada penelitian lanjutan.



Gambar 11. Kurva Training Loss (garis penuh) dan Validation Loss (garis putus-putus) Sembilan Skema Eksperimen: Custom CNN, MobileNetV2, dan InceptionV3 pada Batch Size 8, 16, dan 32

3.7 Pembahasan

Dari perspektif efisiensi komputasi, keunggulan MobileNetV2 dalam penelitian ini sejalan dengan tren pengembangan arsitektur ringkas untuk deteksi penyakit tanaman berbasis mobile, seperti yang dilakukan Mazumder dkk. (2024) [10] yang mengombinasikan DenseNet121 dan MobileNetV2 untuk deteksi penyakit lintas-spesies dengan data terbatas, maupun Zhou dkk. (2024) [11] yang memodifikasi ShuffleNetV2 untuk mengurangi jumlah parameter tanpa mengorbankan akurasi. Meskipun kedua penelitian tersebut menggunakan arsitektur yang berbeda, kesamaan arah temuan ini menegaskan bahwa efisiensi parameter dan akurasi bukanlah dua tujuan yang saling bertentangan, melainkan dapat dicapai secara bersamaan melalui pemilihan arsitektur dan strategi pelatihan yang tepat. Perbandingan ini juga relevan dengan eksplorasi arsitektur berbasis Vision Transformer oleh Han dan Guo (2024) [12], yang meskipun menjanjikan dari sisi akurasi, umumnya memerlukan jumlah parameter dan data pelatihan yang jauh lebih besar dibandingkan arsitektur CNN ringkas seperti MobileNetV2 sebuah pertimbangan penting mengingat keterbatasan infrastruktur komputasi yang umum dijumpai pada konteks pertanian cerdas di negara berkembang seperti Indonesia.

Terkait penanganan ketidakseimbangan kelas, strategi pembobotan kelas berbasis frekuensi terbalik yang diterapkan pada penelitian ini (Subbab 2.5) merupakan bentuk sederhana dari prinsip yang lebih umum diusulkan Cui dkk. (2019) [13] melalui Class-Balanced Loss berbasis effective number of samples, yang memperhitungkan tumpang tindih informasi antar sampel dalam kelas yang sama sehingga penambahan sampel pada kelas besar memberikan kontribusi marjinal yang semakin kecil. Mengingat bobot kelas pada penelitian ini berkisar sangat sempit (0,9083--1,0014) akibat dataset yang telah dikurasi mendekati seimbang, penerapan skema pembobotan yang lebih kompleks seperti effective number of samples kemungkinan belum akan memberikan perbedaan performa yang signifikan pada kondisi dataset saat ini, namun berpotensi menjadi lebih relevan apabila penelitian lanjutan memperluas dataset dengan komoditas atau sumber data baru yang berpotensi menghasilkan ketidakseimbangan kelas yang lebih ekstrem.

Secara keseluruhan, temuan-temuan pada penelitian ini memberikan beberapa implikasi praktis bagi pengembangan sistem identifikasi penyakit tanaman berbasis deep learning di Indonesia. Pertama, pemilihan arsitektur transfer learning yang ringkas seperti MobileNetV2 terbukti lebih menguntungkan dibandingkan arsitektur yang lebih besar seperti InceptionV3, baik dari sisi akurasi maupun kebutuhan sumber daya komputasi, sehingga lebih sesuai untuk diimplementasikan pada aplikasi mobile berbasis smartphone yang umum digunakan petani di lapangan. Kedua, sensitivitas batch size yang berbeda antar arsitektur menegaskan pentingnya melakukan pengujian hyperparameter secara sistematis untuk setiap arsitektur, alih-alih mengasumsikan konfigurasi optimal yang seragam berlaku untuk seluruh model, sejalan dengan rekomendasi tinjauan sistematis Shafik dkk. (2023) [5] mengenai perlunya standarisasi protokol evaluasi pada penelitian deteksi penyakit tanaman. Ketiga, penanganan ketidakseimbangan kelas melalui pembobotan kelas terbukti efektif menjaga performa yang merata antar kelas pada penelitian ini, namun tetap perlu diantisipasi secara lebih ketat pada perluasan penelitian ke dataset yang lebih heterogen di masa depan. Ketiga implikasi ini secara bersama-sama mendukung pengembangan sistem identifikasi penyakit tanaman berbasis deep learning yang tidak hanya akurat, tetapi juga efisien, robust terhadap ketidakseimbangan data, dan interpretable, sebagai bagian dari upaya mewujudkan ketahanan pangan nasional melalui teknologi pertanian cerdas (*smart farming*).

4. KESIMPULAN

Penelitian ini merancang dan membandingkan tiga arsitektur *deep learning* untuk identifikasi 72 kelas penyakit daun tanaman dari 72.000 citra, sekaligus mengevaluasi sensitivitas ketiganya terhadap variasi *batch size*. Hasil pengujian menunjukkan pola sensitivitas *batch size* yang berlawanan arah antara Custom CNN dan kedua model *Transfer Learning*, dengan MobileNetV2 mencapai performa tertinggi (*test accuracy* 97,57%, F1-Score 0,9755) pada *batch size* 8 sekaligus memiliki jumlah parameter paling ringkas ($\pm 2,96$ juta, sekitar 8 kali lebih kecil dibandingkan InceptionV3); seluruh perbedaan performa antar *batch size* pada Custom CNN dan MobileNetV2, serta antara MobileNetV2 dan kedua arsitektur pembanding pada konfigurasi optimalnya, terbukti signifikan secara statistik ($p < 0,05$). Analisis *Explainable AI* (Confusion Matrix, t-SNE, dan Grad-CAM) memperkuat temuan kuantitatif tersebut dengan menunjukkan bahwa



keunggulan MobileNetV2 bersumber dari representasi fitur antarkelas yang lebih terpisah dan atensi model yang terkonsentrasi pada region lesi penyakit, bukan artefak dataset. Berdasarkan kombinasi akurasi tinggi dan ukuran model yang ringkas, MobileNetV2 pada konfigurasi *batch size* 8 merupakan kandidat yang menjanjikan untuk dikembangkan lebih lanjut menuju implementasi pada aplikasi mobile petani dan sistem IoT pertanian; namun demikian, rekomendasi ini bersifat tentatif karena penelitian ini belum mengukur secara langsung waktu inferensi, konsumsi energi, maupun perilaku model pada perangkat *edge/mobile* yang sesungguhnya, sehingga pengukuran tersebut menjadi prasyarat sebelum klaim kelayakan *deployment* dapat dibuat secara definitif. Keterbatasan lain penelitian ini adalah setiap skema hanya dievaluasi pada satu kali proses pelatihan tanpa pengulangan lintas-*seed*, sehingga variabilitas hasil akibat inisialisasi acak belum terkuantifikasi, dan uji signifikansi yang digunakan berbasis proporsi agregat, bukan uji berpasangan pada tingkat prediksi per-sampel. Penelitian lanjutan disarankan untuk: (1) mengukur secara langsung waktu inferensi, konsumsi memori, dan ukuran model pada perangkat mobile/edge sesungguhnya; (2) mengulang pelatihan lintas-*seed* untuk mengkuantifikasi variabilitas hasil; (3) mengeksplorasi teknik augmentasi sintesis berbasis *Generative Adversarial Network* untuk kelas minoritas; (4) menjajaki arsitektur ringkas terkini serta pendekatan berbasis Vision Transformer sebagai pembanding; dan (5) memperluas cakupan dataset dengan komoditas lokal Indonesia agar hasil penelitian lebih merepresentasikan kondisi pertanian nasional secara menyeluruh.

REFERENCES

- [1] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016, doi: 10.3389/fpls.2016.01419.
- [2] P. Palupiningsih, A. R. Sujiwanto, and R. R. B. P. Prawirodirjo, "Analisis Perbandingan Performa Model Klasifikasi Kesehatan Daun Tomat Menggunakan Arsitektur VGG, MobileNet, dan Inception V3," *Jurnal Ilmu Komputer dan Agri-Informatika*, vol. 10, no. 1, pp. 98–110, 2023, doi: 10.29244/jika.10.1.98-110.
- [3] A. Karno, W. Hastomo, I. Wardhana, Sutarno, and D. Arif, "Identifikasi 29 Jenis Penyakit Tanaman Menggunakan Deep Learning EfficientNetB3," *Insearch: Information System Research Journal*, vol. 2, no. 2, pp. 35–45, 2022.
- [4] A. Agustin, M. A. Bianto, and H. Ardiansyah, "Perbandingan Klasifikasi Daun Cabai dengan Metode CNN dan SVM untuk Deteksi Penyakit," *Jurnal Pengembangan Teknologi Informasi dan Komunikasi*, vol. 4, no. 1, pp. 187–196, 2026, doi: 10.52060/juptik.v4i1.4403.
- [5] W. Shafik, A. Tufail, A. Namoun, L. C. De Silva, and R. A. A. H. M. Apong, "A Systematic Literature Review on Plant Disease Detection: Motivations, Classification Techniques, Datasets, Challenges, and Future Trends," *IEEE Access*, vol. 11, pp. 59174–59203, 2023, doi: 10.1109/ACCESS.2023.3284760.
- [6] J. G. A. Barbedo, "Impact of Dataset Size and Variety on the Effectiveness of Deep Learning and Transfer Learning for Plant Disease Classification," *Computers and Electronics in Agriculture*, vol. 153, pp. 46–53, 2018, doi: 10.1016/j.compag.2018.08.013.
- [7] E. A. Aldakheel, M. Zakariah, and A. H. Alabdallal, "Detection and Identification of Plant Leaf Diseases Using YOLOv4," *Frontiers in Plant Science*, vol. 15, 2024, doi: 10.3389/fpls.2024.1355941.
- [8] R. Moyazzoma, M. A. A. Hossain, M. H. Anuz, and A. Sattar, "Transfer Learning Approach for Plant Leaf Disease Detection Using CNN with Pre-Trained Feature Extraction Method MobileNetV2," in *Proc. 2021 2nd Int. Conf. Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2021, pp. 526–529, doi: 10.1109/ICREST51555.2021.9331214.
- [9] L. Li, S. Zhang, and B. Wang, "Plant Disease Detection and Classification by Deep Learning A Review," *IEEE Access*, vol. 9, pp. 56683–56698, 2021, doi: 10.1109/ACCESS.2021.3069646.
- [10] Md. K. A. Mazumder, M. F. Mridha, S. Alfarhood, M. Safran, Md. Abdullah-Al-Jubair, and D. Che, "A Robust and Light-Weight Transfer Learning-Based Architecture for Accurate Detection of Leaf Diseases Across Multiple Plants Using Less Amount of Images," *Frontiers in Plant Science*, vol. 14, 2024, doi: 10.3389/fpls.2023.1321877.
- [11] H. Zhou, J. Chen, X. Niu, Z. Dai, L. Qin, L. Ma, J. Li, Y. Su, and Q. Wu, "Identification of Leaf Diseases in Field Crops Based on Improved ShuffleNetV2," *Frontiers in Plant Science*, vol. 15, 2024, doi: 10.3389/fpls.2024.1342123.
- [12] D. Han and C. Guo, "Automatic Classification of Ligneous Leaf Diseases via Hierarchical Vision Transformer and Transfer Learning," *Frontiers in Plant Science*, vol. 14, 2024, doi: 10.3389/fpls.2023.1328952.
- [13] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-Balanced Loss Based on Effective Number of Samples," in *Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9260–9269, doi: 10.1109/CVPR.2019.00949.
- [14] K. P. Ferentinos, "Deep Learning Models for Plant Disease Detection and Diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.
- [15] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," in *Proc. 5th Int. Conf. Learning Representations (ICLR)*, 2017.
- [16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
- [17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.
- [18] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. 3rd Int. Conf. Learning Representations (ICLR)*, 2015.
- [19] L. van der Maaten and G. Hinton, "Visualizing Data Using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
- [20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *Proc. 2017 IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.