

## Optimasi MobileNetV2 dengan *Pruning* dan *Quantization* untuk Deteksi Penyakit Daun Padi pada Perangkat *Edge*

Nur Dwi Priyambodo<sup>1</sup>, Edi Sugiarto<sup>2\*</sup>

<sup>1,2</sup>Fakultas Ilmu Komputer, Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia

E-mail: <sup>1</sup>nurdwipriyambodo@gmail.com, <sup>2</sup>edi.sugiarto@dsn.dinus.ac.id

(\* : corresponding author)

### Abstrak

Penyakit daun padi Bacterial Leaf Blight (BLB), Brown Spot, dan Leaf Blast merupakan ancaman serius terhadap ketahanan pangan dengan potensi penurunan produksi hingga 30–70%. Model *deep learning* yang ada umumnya berukuran besar (>80 MB) sehingga tidak efisien untuk smartphone petani berspesifikasi rendah. Penelitian ini mengusulkan optimasi MobileNetV2 melalui magnitude-based weight pruning (40%) dan *Post-Training Quantization* (PTQ) Int8 berbasis TensorFlow Lite untuk *deployment* pada perangkat Android. Dataset bersumber dari repositori Mendeley (DOI: 10.17632/fwcj7stb8r/1) berisi 5.932 citra dalam 4 kelas; penelitian ini menggunakan subset 3 kelas (BLB, Brown Spot, Leaf Blast) sebanyak 4.804 citra, dibagi dengan skema stratified 70-15-15 menjadi data latih, validasi, dan uji. Pengujian dilakukan pada 721 citra data uji (*held-out test set*) yang terpisah dari proses pelatihan, validasi, dan kalibrasi kuantisasi. Model yang diusulkan menghasilkan ukuran 2,89 MB (reduksi 88,23% dari *baseline* 24,58 MB), kecepatan inferensi 14,53 ms/gambar pada CPU Intel *workstation*, dan akurasi 98,34% (F1-Score makro 0,9831). Percobaan *pruning* 50% mengakibatkan degradasi akurasi ke 83,50% pasca-*quantization*, mengonfirmasi 40% sebagai konfigurasi *trade-off* terbaik di antara dua tingkat *sparsity* yang diuji. Analisis Grad-CAM mengonfirmasi relevansi biologis fitur yang diekstrak terhadap patologi masing-masing penyakit. Penelitian lanjutan mencakup pengujian *on-device* pada perangkat Android serta perluasan cakupan kelas penyakit.

**Kata kunci:** deteksi penyakit padi, MobileNetV2, *pruning*, *quantization*, *edge computing*

### Abstract

Rice leaf diseases Bacterial Leaf Blight (BLB), Brown Spot, and Leaf Blast pose a serious threat to food security, with potential yield losses of 30–70%. Existing deep learning models are typically large (>80 MB), making them impractical for farmers' low-specification smartphones. This study proposes MobileNetV2 optimization through 40% magnitude-based weight pruning and Post-Training Quantization (PTQ) Int8 using TensorFlow Lite for Android deployment. The dataset is sourced from a Mendeley repository (DOI: 10.17632/fwcj7stb8r/1) comprising 5,932 images across 4 classes, from which a 3-class subset (BLB, Brown Spot, Leaf Blast) totaling 4,804 images was used, split 70-15-15 into training, validation, and test sets. Evaluation was performed on 721 held-out test images, excluded from the training, validation, and quantization calibration processes. The proposed model achieves 2.89 MB (88.23% reduction from the 24.58 MB baseline), an inference speed of 14.53 ms/image on an Intel CPU workstation, and an accuracy of 98.34% (macro F1-Score 0.9831). More aggressive 50% pruning caused accuracy to degrade to 83.50% post-quantization, confirming 40% as the best trade-off among the sparsity levels tested. Grad-CAM analysis validated the biological relevance of the extracted features for each disease class. Future work includes on-device validation on Android hardware and expanded disease class coverage.

**Keywords:** rice disease detection, MobileNetV2, *pruning*, *quantization*, *edge computing*

## 1. PENDAHULUAN

Padi (*Oryza sativa* L.) merupakan komoditas pangan strategis yang menjadi sumber kalori utama bagi lebih dari separuh populasi dunia, khususnya di kawasan Asia Tenggara [1]. Di Indonesia, kehilangan hasil panen akibat penyakit daun padi dapat mencapai 30–70% dari total produksi, dengan Bacterial Leaf Blight (BLB), Brown Spot, dan Leaf Blast sebagai penyebab utama [1], [2], [23]. Paradigma *edge computing* pemrosesan data langsung pada perangkat lokal tanpa bergantung pada server *cloud* menjadi kunci penerapan kecerdasan buatan di bidang pertanian, terutama di wilayah dengan keterbatasan koneksi internet.

Pendekatan *deep learning* berbasis CNN telah terbukti mampu mendeteksi penyakit daun tanaman dengan akurasi tinggi [3], [4], [5], [6], [7]. namun *deployment*-nya di tingkat petani

masih terkendala ukuran model yang besar dan kecepatan inferensi yang rendah pada smartphone berspesifikasi terbatas. [24] Rentang ukuran 80–150 MB yang umum dijumpai pada model deteksi penyakit tanaman tercermin dari beberapa studi terdahulu: Bari et al. [14] melaporkan model >50 MB, Latif et al. [4] ~83 MB, dan Paymode & Malode [15] >100 MB. MobileNetV2 [9] dirancang untuk mengatasi keterbatasan ini melalui *depthwise separable convolution* dan *inverted residual blocks*, namun model *baseline*-nya tetap berukuran ~26 MB sehingga kompresi lanjutan masih diperlukan untuk *deployment* pada perangkat mobile berspesifikasi rendah.

Tiga *research gap* teridentifikasi dari kajian literatur: (1) model dengan akurasi tinggi umumnya tidak mempertimbangkan efisiensi *deployment* pada perangkat *edge*; (2) kombinasi *magnitude-based pruning* dan *Post-Training Quantization* (PTQ) Int8 secara simultan pada MobileNetV2 untuk klasifikasi tiga kelas penyakit daun padi (BLB, Brown Spot, Leaf Blast) belum dieksplorasi secara komprehensif; (3) validasi inferensi TFLite pada data uji berskala besar dengan metrik evaluasi yang lengkap masih terbatas [10], [11]. Penelitian ini mengisi ketiga gap tersebut melalui analisis komparatif sistematis atas empat konfigurasi model. Perbedaan penelitian ini dengan riset terdekat ditegaskan sebagai berikut: Nugroho et al. [10] menerapkan MobileNetV2 dan *quantization* pada platform mikrokontroler STM32, sedangkan penelitian ini menyoroti ekosistem Android yang secara arsitektural dan sumber daya berbeda; Martins et al. [11] menerapkan *quantization* tanpa tahap *pruning*, sehingga potensi kompresi ukuran tambahan dari pemangkasan bobot belum tereksplorasi; Karim et al. [17] menggabungkan *pruning* ringan dan Grad-CAM pada perangkat *edge* nyata, namun pada objek tanaman anggur dengan karakteristik lesi yang berbeda dari penyakit daun padi. Dengan demikian, kebaruan penelitian ini terletak pada kombinasi simultan *pruning* dan PTQ Int8 pada MobileNetV2 yang dievaluasi secara lengkap (*accuracy*, *precision*, *recall*, F1-Score, ukuran model, dan *inference time*) pada data uji berskala penuh (721 citra) untuk kasus tiga kelas penyakit daun padi yang diarahkan pada *deployment* Android — kombinasi yang belum ditemukan secara utuh pada studi-studi sebelumnya.

Kontribusi penelitian ini meliputi: (1) analisis komparatif *pruning* 50% vs. 40% yang mengindikasikan ambang *sparsity* yang lebih unggul di antara kedua konfigurasi yang diuji pada MobileNetV2 untuk kasus dataset ini; (2) bukti empiris bahwa kombinasi *pruning* 40% dan PTQ Int8 menghasilkan akurasi 98,34% pada ukuran model 2,89 MB; (3) validasi interpretabilitas model melalui Grad-CAM [12] yang mengonfirmasi korespondensi antara fitur yang diekstraksi CNN dan ciri patologi visual masing-masing kelas penyakit.

Studi terdahulu umumnya menempatkan akurasi dan efisiensi ukuran model sebagai *trade-off* yang saling eksklusif. Prajapati et al. [13] mencapai 82,31% dengan SVM berbasis fitur GLCM, tanpa kemampuan *edge deployment*. Bari et al. [14] mencapai 93,75% dengan Faster R-CNN, tetapi ukuran model >50 MB tidak dioptimasi untuk perangkat mobile. Latif et al. [4] mencapai 97,50% dengan *improved CNN* pada ukuran ~83 MB, dan Paymode & Malode [15] mencapai 98,40% dengan *transfer learning* VGG pada ukuran >100 MB keduanya melampaui kapasitas penyimpanan smartphone petani berspesifikasi rendah. Karim et al. [17] mencapai 99,42% menggunakan MobileNetV3 dan Grad-CAM pada *edge device* nyata, namun pada objek tanaman anggur, bukan padi. Nugroho et al. [10] paling mendekati topik penelitian ini dengan kombinasi MobileNetV2 dan *quantization* pada MCU STM32 (akurasi 99%, ~4 MB), tetapi platform mikrokontroler tersebut berbeda secara signifikan dari ekosistem Android yang menjadi target *deployment* pada penelitian ini. Martins et al. [11] mengusulkan *quantized CNN* dengan akurasi 96,40% (~8 MB) tanpa *pruning*. Dari kajian ini, teridentifikasi bahwa kombinasi *magnitude-based pruning* dan PTQ Int8 secara simultan pada MobileNetV2 untuk klasifikasi tiga kelas penyakit daun padi dan *deployment* Android belum dieksplorasi secara komprehensif. Perlu ditekankan bahwa dataset yang digunakan dalam penelitian ini bersumber dari repositori terbuka (Mendeley Data) dan merepresentasikan kondisi citra daun padi secara umum, bukan kondisi lapangan spesifik Indonesia; validasi pada data lapangan Indonesia menjadi salah satu arah penelitian lanjutan.

Secara teoretis, fondasi *magnitude-based pruning* diletakkan oleh Han et al. [20], yang membuktikan bahwa bobot bermagnitudo kecil dapat dieliminasi tanpa degradasi akurasi yang

signifikan. Jacob et al. [21] kemudian memformalisasi *quantization* integer penuh yang memungkinkan inferensi efisien pada perangkat keras terbatas. Kombinasi kedua teknik inilah yang menjadi inti metodologi penelitian ini untuk menghasilkan model kompak yang dapat berjalan pada smartphone petani.

## 2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental kuantitatif komparatif dengan mengevaluasi empat konfigurasi model secara sistematis: (1) *baseline* MobileNetV2, (2) *pruning* 50%, (3) *pruning* 50% + PTQ Int8, dan (4) *pruning* 40% + PTQ Int8 (model yang diusulkan). Lingkungan pengembangan menggunakan Python 3.10, TensorFlow 2.x, dan TensorFlow Model Optimization Toolkit pada *workstation* dengan environment Miniconda (rice\_ai). Alur penelitian secara keseluruhan ditunjukkan pada Gambar 2.

### 2.1. Dataset

Dataset primer adalah *Rice Leaf Image Samples* dari Mendeley Data, Version 1 (DOI: 10.17632/fwcj7stb8r/1), dipublikasikan 18 Juli 2020 oleh Sethy et al. [22], dengan rujukan publikasi pendamping pada jurnal Computers and Electronics in Agriculture [22]. Dataset ini berisi 5.932 citra dalam 4 kelas (Bacterial Leaf Blight 1.584, Blast 1.440, Brown Spot 1.600, dan Tungro 1.308 citra). Metadata repositori tidak melaporkan secara eksplisit periode/tanggal akuisisi citra di lapangan; informasi ini merupakan keterbatasan metadata pada sumber dataset asli dan bukan pada prosedur penelitian ini (lihat Bagian Keterbatasan). Seluruh eksperimen dalam penelitian ini menggunakan Version 1 dataset sebagaimana tercantum pada DOI di atas; pembaca yang ingin mereproduksi hasil perlu memastikan mengakses versi yang sama, mengingat repositori ini memiliki Version 2 (2024) yang dipublikasikan terpisah dengan kemungkinan perubahan komposisi data. Penelitian ini hanya menggunakan 3 kelas yang relevan dengan penyakit padi target, yaitu BLB, Brown Spot, dan Leaf Blast, dengan total 4.804 citra (BLB 1.644, Brown Spot 1.660, Leaf Blast 1.500); kelas Tungro (1.308 citra) tidak disertakan sejak tahap penyusunan folder mentah sehingga tidak pernah masuk ke proses *splitting*. Secara klinis, ketiga kelas target dapat dibedakan melalui ciri visual berikut: BLB (disebabkan bakteri *Xanthomonas oryzae*) ditandai lesi memanjang berwarna kuning-kecokelatan yang bermula dari tepi daun dan meluas mengikuti tulang daun; Brown Spot (disebabkan jamur *Bipolaris oryzae*) ditandai bercak oval berwarna cokelat gelap dengan bagian tengah lebih terang membentuk pola eye-spot yang tersebar di permukaan daun; Leaf Blast (disebabkan jamur *Magnaporthe oryzae*) ditandai lesi berbentuk belah ketupat (rhombus/spindle-shaped) dengan bagian tengah abu-abu keputihan dan tepi cokelat kemerahan. Contoh citra representatif masing-masing kelas ditampilkan pada Gambar 1a. Distribusi kelas relatif seimbang dengan proporsi BLB 34,2%, Brown Spot 34,6%, dan Leaf Blast 31,2%. Pembagian stratified 70-15-15 menggunakan *sklearn.model\_selection.train\_test\_split* (random\_state=42) menghasilkan Train=3.362, Val=721, dan Test=721 citra ( $3.362 + 721 + 721 = 4.804$ ), konsisten dengan total citra tiga kelas di atas. *Validation set* digunakan hanya untuk pemantauan kinerja selama pelatihan dan seleksi model terbaik via *EarlyStopping*; *test set* dijaga sebagai *held-out set* yang tidak digunakan dalam proses *training*, validasi, kalibrasi *pruning*, kalibrasi kuantisasi, maupun pemilihan *hyperparameter*.



Gambar 1a. Contoh Citra Representatif Setiap Kelas Penyakit: (a) Bacterial Leaf Blight, (b) Brown Spot, (c) Leaf Blast

Pra-pemrosesan meliputi *resize* 224×224 piksel, normalisasi 1/255, dan augmentasi khusus *training* (*rotation* 30°, *zoom* 0.2, *flip* horizontal/vertikal, *brightness* [0.8, 1.2]). *Validation* dan *test set* hanya mengalami *rescaling* tanpa augmentasi.

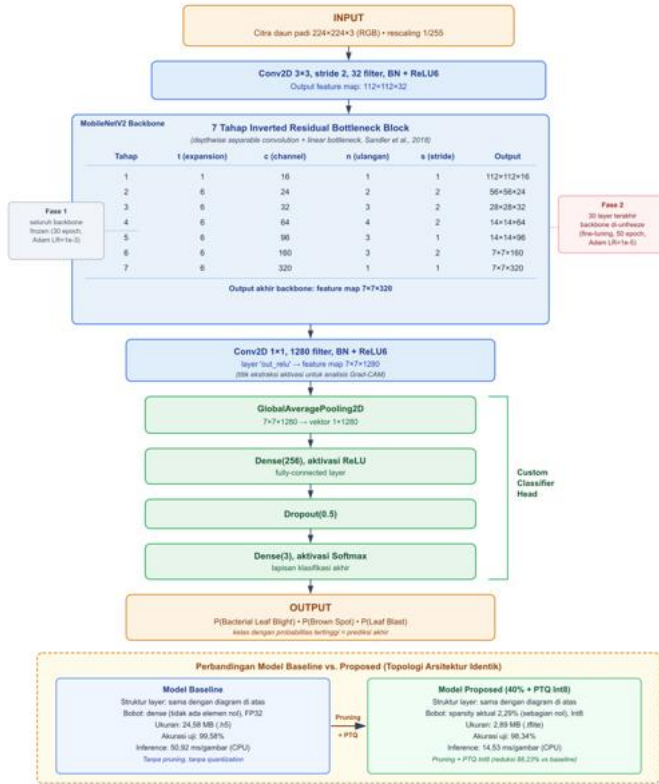
Ketersediaan Dataset. Dataset yang digunakan bersifat open access dan dapat diakses publik melalui repositori Mendeley Data pada tautan <https://doi.org/10.17632/fwcj7stb8r.1>, dipublikasikan oleh Sethy et al. (2020) di bawah lisensi *Creative Commons Attribution* 4.0 (CC BY 4.0) sehingga bebas diunduh, digunakan, dan didistribusikan ulang dengan mencantumkan atribusi yang sesuai. Perlu ditegaskan bahwa seluruh angka dan pembagian data pada penelitian ini merujuk pada Version 1 dataset (DOI: 10.17632/fwcj7stb8r.1, dipublikasikan 18 Juli 2020), bukan Version 2 yang dipublikasikan Mendeley pada 2024 dengan DOI terpisah. Subset tiga kelas yang digunakan (Bacterial Leaf Blight, Brown Spot, Leaf Blast; total 4.804 citra) beserta skema pembagian *train/validation/test* 70:15:15 dapat direproduksi langsung dari sumber tersebut menggunakan *sklearn.model\_selection.train\_test\_split* dengan *random\_state*=42 sebagaimana dijelaskan di atas.

## 2.2. Arsitektur dan Pelatihan Baseline

Model menggunakan MobileNetV2 pre-trained ImageNet dengan custom classifier head: GlobalAveragePooling2D → Dense(256, ReLU) → Dropout(0.5) → Dense(3, Softmax). Ukuran Dense(256) dipilih berdasarkan praktik terbaik yang umum untuk classifier head pada MobileNetV2 fine-tuning nilai ini memberikan kapasitas representasi yang cukup tanpa menambahkan kompleksitas berlebih [22]. Pelatihan dilakukan dua fase: Phase 1 (30 epoch, base frozen, Adam LR=1e-3) dan Phase 2 fine-tuning (50 epoch, 30 layer terakhir dibuka, Adam LR=1e-5). Strategi augmentasi (*rotation*, *zoom*, *flip*, *brightness*) mengikuti praktik standar yang terbukti meningkatkan generalisasi pada dataset citra daun tanaman [4], [15]. Model *baseline* mencapai akurasi test 99% dengan ukuran file .h5 sebesar 26 MB.

Secara lebih rinci, tahapan arsitektur CNN yang digunakan diilustrasikan pada Gambar 1. Input berupa citra 224×224×3 piksel (hasil *rescaling* 1/255) diproses melalui lapisan stem Conv2D 3×3 (stride 2, 32 filter, BN + ReLU6) menghasilkan feature map 112×112×32. Feature map tersebut selanjutnya melewati tujuh tahap inverted residual *bottleneck block* khas MobileNetV2 (*depthwise separable convolution dengan linear bottleneck*), dengan konfigurasi berurutan: tahap 1 (t=1, c=16, n=1, s=1), tahap 2 (t=6, c=24, n=2, s=2), tahap 3 (t=6, c=32, n=3, s=2), tahap 4 (t=6, c=64, n=4, s=2), tahap 5 (t=6, c=96, n=3, s=1), tahap 6 (t=6, c=160, n=3, s=2), dan tahap 7 (t=6, c=320, n=1, s=1), sehingga dihasilkan feature map akhir berukuran 7×7×320. Feature map ini kemudian diproses oleh lapisan Conv2D 1×1 (1280 filter, BN + ReLU6) yang dikenal sebagai layer 'out\_relu', menghasilkan feature map 7×7×1280 lapisan inilah yang menjadi titik ekstraksi aktivasi untuk analisis Grad-CAM pada tahap evaluasi. Selanjutnya, *classifier head* kustom mengambil alih proses klasifikasi: GlobalAveragePooling2D meringkas feature map 7×7×1280 menjadi vektor tunggal berukuran 1×1280, dilanjutkan lapisan Dense(256) dengan aktivasi ReLU sebagai fully-connected layer, Dropout(0.5) untuk regularisasi guna mencegah overfitting, dan lapisan Dense(3) dengan aktivasi Softmax sebagai klasifikasi akhir yang menghasilkan probabilitas untuk tiga kelas penyakit (Bacterial Leaf Blight, Brown Spot, dan Leaf Blast). Sejalan dengan strategi dua fase yang telah dijelaskan, Perlu ditegaskan bahwa topologi arsitektur yang diilustrasikan pada Gambar 1 berlaku identik untuk model baseline maupun model hasil pruning (40% dan 50%): kedua tahap kompresi tersebut tidak mengubah jumlah maupun urutan layer, melainkan hanya memodifikasi nilai bobot (pruning, dengan mengonversi sebagian bobot bermagnitudo kecil menjadi nol) dan representasi numerik bobot serta aktivasi (*Post-Training Quantization* Int8, dari format *floating-point* 32-bit menjadi integer 8-bit). Dengan demikian, perbedaan antara model *baseline* dan model *proposed* (40%+PTQ Int8) bersifat internal pada level parameter, bukan pada struktur arsitektur CNN yang tervisualisasi. Model baseline menggunakan seluruh arsitektur pada Gambar 1 dengan bobot dense (tidak ada elemen nol) dalam format *floating-point* 32-bit, sedangkan model *proposed* menggunakan arsitektur yang sama dengan sebagian kecil bobot bernilai nol hasil pruning (sparsity aktual 2,29% pada konfigurasi 40%, lihat Bagian 3.2) dan seluruh bobot serta aktivasi telah dikuantisasi ke format

integer 8-bit. Fase 1 membekukan seluruh parameter backbone MobileNetV2 sehingga hanya classifier head yang dilatih selama 30 epoch, sedangkan Fase 2 membuka 30 layer terakhir backbone (mencakup tahap bottleneck akhir dan lapisan konvolusi 1×1 'out\_relu') untuk proses fine-tuning selama 50 epoch dengan learning rate yang lebih rendah agar representasi fitur hasil pre-training pada ImageNet tidak rusak (*catastrophic forgetting*).



Gambar 1. Detail Tahapan Arsitektur CNN: Backbone MobileNetV2 (7 Tahap Bottleneck Block) dan Custom Classifier Head. Arsitektur ini berlaku untuk model baseline maupun model hasil pruning+PTQ Int8 (proposed); perbedaan kedua model terletak pada nilai bobot dan representasi numerik parameter, bukan pada struktur layer yang ditampilkan.

2.3. Magnitude-Based Pruning

Magnitude-based weight *pruning* diterapkan menggunakan TensorFlow Model Optimization Toolkit (TF-MOT) [20] dengan PolynomialDecay schedule. Metode ini didasarkan pada prinsip magnitude-based weight removal yang dirintis oleh Han et al. [20], di mana bobot dengan magnitudo terkecil dieliminasi secara iteratif karena berkontribusi minimal terhadap prediksi. Konfigurasi schedule dipilih sebagai berikut: *initial\_sparsity*=0 (*pruning* dimulai dari model penuh untuk stabilitas awal), *end\_step*=10.000 (memberikan cukup iterasi untuk konvergensi bertahap sesuai rekomendasi dokumentasi TF-MOT), *LR*=1e-6 (learning rate rendah untuk *fine-tuning* pasca-*pruning* tanpa mengganggu representasi yang telah dipelajari), dan *EarlyStopping patience*=5. Penetapan dua level *sparsity* yang diuji, yaitu 40% dan 50%, merupakan bagian dari desain eksperimen komparatif untuk mengeksplorasi ambang batas kompresi terhadap kinerja model, bukan penentuan nilai optimal yang ditetapkan di awal. Kedua konfigurasi ini dipilih untuk merepresentasikan tingkat kompresi sedang (40%) dan agresif (50%), dengan penentuan konfigurasi yang lebih unggul dilakukan berdasarkan hasil evaluasi komparatif yang disajikan pada Bagian 3. Dua konfigurasi diuji: *pruning* 50% (*final\_sparsity*=0.50) dan *pruning* 40% (*final\_sparsity*=0.40). *Strip\_pruning()* diterapkan setelah konvergensi.

2.4. Post-Training Quantization (PTQ Int8)

Tahap akhir kompresi model dilakukan melalui Post-Training Quantization (PTQ) Int8,

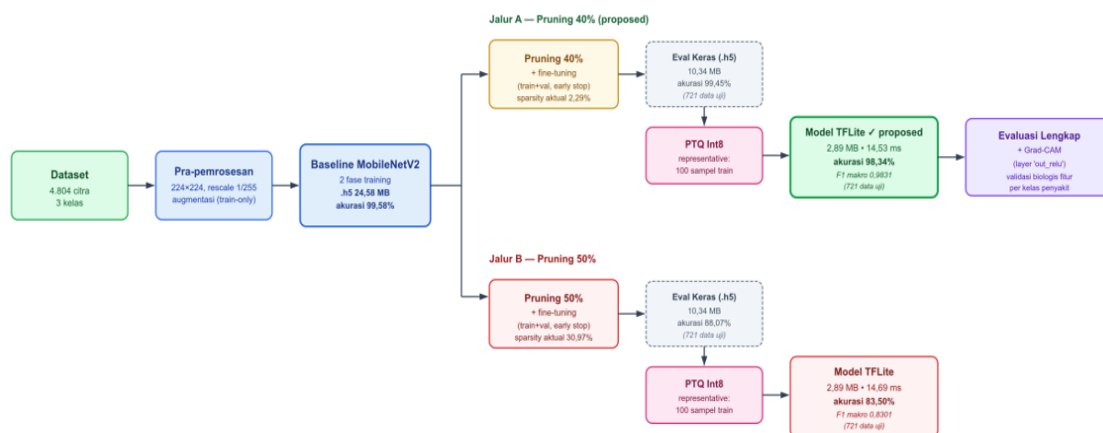
yang bertujuan mengonversi bobot model dari representasi floating-point 32-bit menjadi representasi integer 8-bit sehingga diperoleh model dengan ukuran lebih kecil dan latensi inferensi lebih rendah tanpa memerlukan pelatihan ulang penuh. PTQ Int8 diterapkan pada kedua model *pruned* menggunakan *TFLiteConverter* dengan *full integer quantization* [21]: *converter.optimizations* = [tf.lite.Optimize.DEFAULT], *representative\_dataset*, dan *target\_spec.supported\_ops* = [tf.lite.OpsSet.TFLITE\_BUILTINS\_INT8]. Parameter *representative\_dataset* diperlukan pada tahap ini karena skema full integer quantization mengharuskan konverter mengetahui terlebih dahulu rentang nilai aktivasi setiap layer melalui sejumlah sampel data riil, sebelum rentang tersebut dipetakan ke domain integer 8-bit. *Representative dataset* untuk kalibrasi kuantisasi diambil secara eksklusif dari data latih (*training set*) sebanyak 100 sampel acak (*batch size*=1), dan tidak pernah menyertakan data validasi maupun data uji, guna menjamin bahwa proses kalibrasi tidak mengakses maupun bocor ke *held-out test set*. Karena prosedur kalibrasi ini diterapkan secara identik pada kedua konfigurasi pruning (40% dan 50%), perbedaan performa yang muncul pasca-kuantisasi dapat diatribusikan pada karakteristik masing-masing model *pruned*, bukan pada perbedaan perlakuan saat kalibrasi. Kedua model menghasilkan ukuran TFLite identik: 2,89 MB.

## 2.5. Evaluasi dan Grad-CAM

Evaluasi TFLite menggunakan loop inferensi indexing eksplisit pada seluruh 721 sampel uji. Metrik meliputi: Accuracy, Precision, Recall, F1-Score (per kelas dan macro average), Inference Speed, dan Model Size. Analisis Grad-CAM [12], [25] dilakukan pada layer 'out\_relu' (feature map 7×7, sebelum Global Average Pooling) menggunakan GradientTape.

## 2.6. Pipeline Metodologi

Gambar 2 mengilustrasikan pipeline metodologi penelitian ini secara keseluruhan. Alur terdiri dari enam tahap berurutan: (1) Pengumpulan Dataset dari Mendeley Data dan seleksi tiga kelas penyakit; (2) Pra-pemrosesan yang meliputi resize, normalisasi, dan augmentasi khusus training; (3) Pelatihan *baseline* MobileNetV2 dua fase dengan *transfer learning*; (4) *Magnitude-Based pruning* menggunakan TF-MOT dengan dua konfigurasi sparsity (40% dan 50%); (5) *Post-training Quantization* Int8 via TFLiteConverter untuk konversi ke format TFLite; dan (6) Evaluasi komprehensif yang mencakup metrik klasifikasi, profiling inference time dan model size, serta analisis interpretabilitas Grad-CAM. TensorFlow Lite (TFLite) adalah format model teroptimasi yang dikembangkan oleh Google khusus untuk inferensi deep learning pada perangkat dengan sumber daya terbatas (mobile dan embedded), yang mendukung representasi bobot berpresisi rendah (misalnya Int8) sehingga menghasilkan ukuran model lebih kecil dan latensi inferensi lebih rendah dibandingkan format model standar (.h5/SavedModel) [21]. Pipeline ini dirancang untuk menghasilkan model TFLite yang siap dideploy pada perangkat Android.



Gambar 2. Pipeline Metodologi Penelitian: Dataset → Pra-pemrosesan → Pelatihan *baseline* → *pruning* (40%/50%) → PTQ Int8 → Evaluasi & Grad-CAM Tahap (1) Dataset: pengumpulan 4.804 citra dari repositori Mendeley dan seleksi tiga kelas penyakit target (BLB, Brown Spot, Leaf Blast); Tahap (2) Pra-pemrosesan: resize 224×224 piksel,

normalisasi 1/255, dan augmentasi (rotasi, zoom, flip, brightness) khusus data latih; Tahap (3) Pelatihan Baseline: fine-tuning MobileNetV2 pre-trained ImageNet dua fase menghasilkan model .h5 dengan akurasi 99,58%; Tahap (4) Pruning: pemangkasan bobot magnitude-based via TF-MOT pada dua konfigurasi sparsity nominal (40% dan 50%) menghasilkan model .h5 terpangkas; Tahap (5) PTQ Int8: konversi model .h5 menjadi format TFLite terkuantisasi Int8 menggunakan 100 sampel data latih sebagai representative dataset; Tahap (6) Evaluasi & Grad-CAM: pengujian model TFLite pada 721 data uji held-out mencakup akurasi, precision, recall, F1-Score, ukuran model, waktu inferensi, serta analisis interpretabilitas Grad-CAM pada layer 'out\_relu'.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Posisi Model *Proposed* terhadap Penelitian Terdahulu

Tabel 1 menyajikan perbandingan model *proposed* dengan sembilan penelitian terdahulu. Sebagaimana telah dibahas pada Bagian 1, penelitian yang ada umumnya memilih antara akurasi tinggi (dengan ukuran besar) atau ukuran kecil (dengan akurasi lebih rendah). Model *proposed* mencapai akurasi 98,34% pada ukuran 2,89 MB, menempatkannya pada posisi yang kompetitif di antara sembilan studi pembanding, yakni akurasi tertinggi kedua pada ukuran model terkecil. Perbandingan ini bersifat tidak sepenuhnya *apple-to-apple* karena setiap studi menggunakan dataset, perangkat keras, dan skenario pengujian yang berbeda; karena itu istilah *Pareto-optimal* tidak digunakan secara formal dalam penelitian ini, dan posisi model proposed dimaknai sebagai *trade-off* akurasi–ukuran yang kompetitif dalam konteks sembilan studi yang diperbandingkan, bukan sebagai optimalitas Pareto pada ruang pencarian yang terkontrol. Perbandingan pada *benchmark* standar dengan protokol pengujian yang seragam diperlukan pada penelitian lanjutan untuk memvalidasi klaim ini secara lebih ketat.

Tabel 1. Perbandingan Penelitian Terdahulu dan Posisi Penelitian Ini

| No. | Penulis & Tahun              | Metode                                   | Akurasi | Ukuran  | Gap / Keterbatasan                                                                                                                                |
|-----|------------------------------|------------------------------------------|---------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | Prajapati et al. (2017) [13] | SVM + GLCM                               | 82,31%  | N/A     | Tanpa deep learning; tanpa edge device                                                                                                            |
| 2   | Bari et al. (2021) [14]      | CNN + SVM                                | 93,75%  | >50 MB  | Model sangat besar; tanpa kompresi edge; bukan TFLite                                                                                             |
| 3   | Latif et al. (2022) [4]      | Improved CNN                             | 97,50%  | ~83 MB  | Terlalu besar untuk smartphone petani                                                                                                             |
| 4   | Paymode & Malode (2022) [15] | Transfer Learning VGG                    | 98,40%  | >100 MB | Multi-crop; tidak ada kompresi model                                                                                                              |
| 5   | Haridasan et al. (2023) [16] | Deep Learning CNN                        | 97,2%   | ~50 MB  | Tidak ada optimasi edge deployment                                                                                                                |
| 6   | Karim et al. (2024) [17]     | MobileNetV3+Grad-CAM                     | 99,42%  | ~15 MB  | Tanaman anggur; bukan padi; bukan Android                                                                                                         |
| 7   | Nugroho et al. (2024) [10]   | MobileNetV2+ <i>Quantization</i>         | 99,00%  | ~4 MB   | Platform MCU STM32; bukan Android                                                                                                                 |
| 8   | Martins et al. (2025) [11]   | Quantized CNN                            | 96,40%  | ~8 MB   | Tanpa <i>pruning</i> ; dataset lab-controlled                                                                                                     |
| 9   | Penelitian Ini               | MobileNetV2+ <i>pruning</i> 40%+PTQ Int8 | 98,34%  | 2,89 MB | <i>Trade-off</i> akurasi–ukuran yang kompetitif dalam sembilan studi ini; belum diverifikasi pada <i>benchmark</i> standar dan perangkat Android. |

#### 3.2. Hasil Komparatif Empat Konfigurasi Model

Tabel 2 menyajikan perbandingan seluruh konfigurasi model, dievaluasi secara konsisten pada 721 data uji *held-out* yang sama untuk setiap tahap (Keras .h5 dan TFLite Int8), sehingga tidak terdapat pencampuran antara metrik validasi dan metrik uji. Model proposed (*pruning* 40% + PTQ Int8) mencapai akurasi uji 98,34% dengan ukuran 2,89 MB, reduksi 88,23% dari

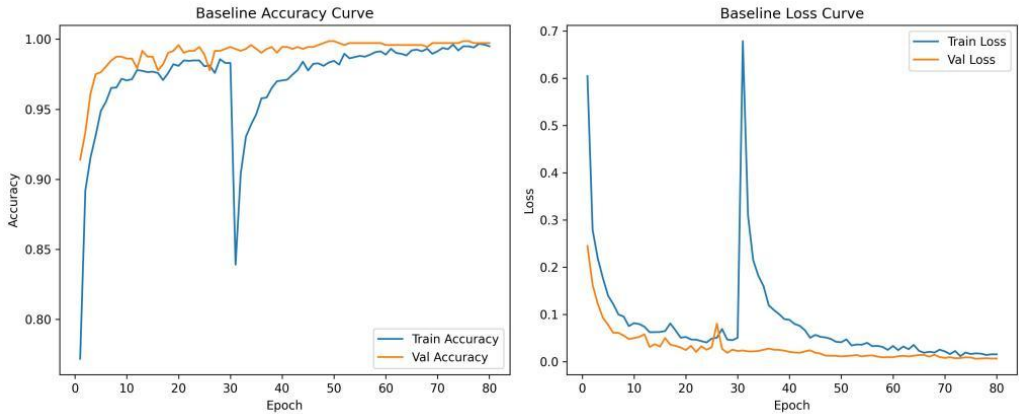
*baseline .h5* (24,58 MB). Dibandingkan *pruning* 50%+PTQ yang mencapai akurasi uji 83,50%, konfigurasi 40% menunjukkan keunggulan signifikan. Namun demikian, pemeriksaan ulang terhadap *sparsity* aktual pada bobot model tersimpan menunjukkan bahwa model 40% hanya mencapai *sparsity* aktual 2,29% (bukan 40% nominal), sedangkan model 50% mencapai *sparsity* aktual 30,97% — mendekati target nominalnya. Hal ini terjadi karena kombinasi *end\_step* penjadwalan *PolynomialDecay* (10.000 langkah) yang jauh melebihi jumlah langkah efektif *training* pasca-*early stopping* (636–1.166 langkah), sehingga sebagian besar konfigurasi *pruning* tidak mencapai target *sparsity* yang dimaksudkan. Implikasinya, keunggulan akurasi konfigurasi 40% tidak murni membuktikan superioritas *sparsity ratio* 40% dibanding 50%, melainkan sebagian besar mencerminkan bahwa model 40% secara efektif hampir tidak mengalami pemangkasan bobot yang substansial. Kesimpulan ini dijelaskan lebih lanjut sebagai keterbatasan pada Bagian 3.5.

*Inference time* 14,53 ms/gambar diukur pada CPU Intel *workstation* menggunakan *runtime* XNNPACK dan belum merepresentasikan performa pada perangkat Android nyata; validasi *on-device* diperlukan untuk mengonfirmasi kelayakan waktu tanggap pada perangkat sasaran (lihat Bagian 3.5).

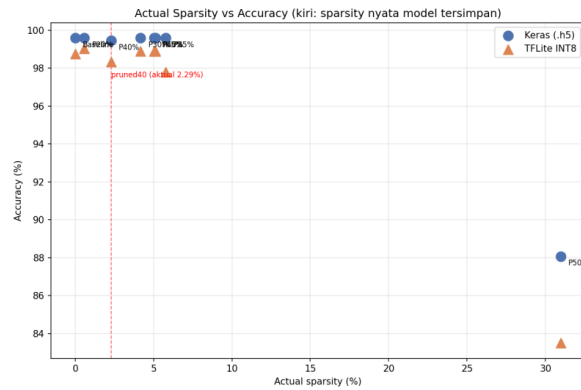
Tabel 2 Hasil Eksperimen Komparatif seluruh konfigurasi dievaluasi pada 721 data uji (test set)

| Metrik                                 | Baseline | Pruning 50% (Keras) | 50%+PTQ Int8 | Pruning 40% (Keras) | 40%+PTQ Int8 ✓ (proposed) |
|----------------------------------------|----------|---------------------|--------------|---------------------|---------------------------|
| Target sparsity                        | 0%       | 50%                 | 50%          | 40%                 | 40%                       |
| Sparsity aktual                        | 0,00%    | 30,97%              | 30,97%       | 2,29%               | 2,29%                     |
| Akurasi uji (%)                        | 99,58    | 88,07               | 83,50        | 99,45               | 98,34                     |
| Precision (macro)                      | 0,9959   | 0,8799              | 0,8398       | 0,9945              | 0,9840                    |
| Recall (macro)                         | 0,9957   | 0,8789              | 0,8304       | 0,9943              | 0,9825                    |
| F1-Score (macro)                       | 0,9958   | 0,8792              | 0,8301       | 0,9944              | 0,9830                    |
| Ukuran Model (MB)                      | 24,58    | 10,34               | 2,89         | 10,34               | 2,89                      |
| Inference Time (ms, CPU)               | 50,92    | 60,93               | 14,69        | 62,26               | 14,53                     |
| Kompresi Ukuran vs <i>baseline .h5</i> | —        | 57,91%              | 88,23%       | 57,91%              | 88,23%                    |

Gambar 3 menunjukkan spike loss di epoch ke-30 akibat transisi Phase-1→Phase-2 yang normal. Gambar 4 menampilkan *scatter plot trade-off* akurasi–ukuran keempat konfigurasi pada data uji yang konsisten, memperlihatkan posisi model *proposed* (40%+PTQ) pada kuadran akurasi tinggi dengan ukuran terkecil di antara model-model terkompresi dalam penelitian ini.



Gambar 3. Kurva Training *baseline* MobileNetV2 (Akurasi dan Loss, 80 Epoch)

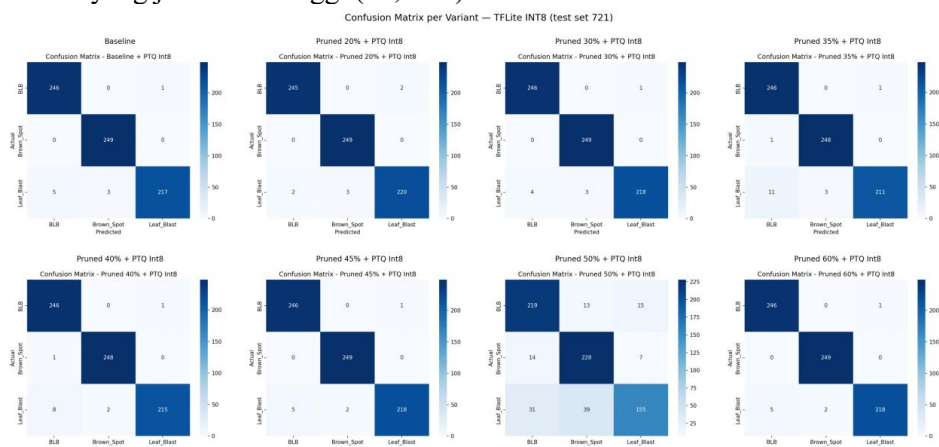


Gambar 4. Sparsity Aktual vs Akurasi pada Seluruh Varian Model (Keras .h5 dan TFLite Int8, N=721). Sumbu horizontal (x) menunjukkan persentase sparsity aktual pada bobot model tersimpan, bukan target sparsity nominal, sedangkan sumbu vertikal (y) menunjukkan akurasi klasifikasi pada 721 data uji. Penanda lingkaran (●) merepresentasikan model Keras (.h5) dan penanda segitiga (▲) merepresentasikan model TFLite Int8 hasil kuantisasi dari model Keras yang bersesuaian. Garis vertikal putus-putus menandai posisi sparsity aktual model pruned40 (2,29%), mengilustrasikan kesenjangan antara target nominal (40%) dan kondisi bobot yang sesungguhnya tersimpan. Grafik ini secara keseluruhan memperlihatkan bahwa akurasi relatif stabil pada rentang sparsity aktual rendah (0–6%) dan baru mengalami penurunan tajam pada sparsity aktual tertinggi (30,97%, varian pruned50), baik pada model Keras maupun TFLite Int8.

### 3.2 Analisis Per-Kelas dan Confusion Matrix

Tabel 3 menyajikan classification report per-kelas model proposed. Brown Spot mencapai F1-Score tertinggi (0,9940) di antara tiga kelas, konsisten dengan pola *eye-spot* konsentris yang relatif mudah dibedakan secara visual. BLB memperoleh F1-Score 0,9802 dengan recall tinggi (0,9960) namun precision sedikit lebih rendah (0,9647) akibat 9 sampel dari kelas lain yang salah diklasifikasikan sebagai BLB. Leaf Blast menunjukkan recall terendah (0,9556) dengan 10 *false negative* (8 diklasifikasikan sebagai BLB, 2 sebagai Brown Spot), mengindikasikan kesulitan model pada citra dengan lesi awal yang pola *rhombus*-nya belum terbentuk sempurna.

Confusion matrix seluruh varian, termasuk model proposed, ditampilkan pada Gambar 5. Grid tersebut mengonfirmasi bahwa kesalahan klasifikasi pada model proposed (40%+PTQ) terkonsentrasi pada kelas Leaf Blast (8 sampel diklasifikasikan sebagai BLB, 2 sebagai Brown Spot), sementara BLB dan Brown Spot terklasifikasi hampir sempurna — pola yang konsisten di seluruh varian kecuali pruned50, yang menunjukkan degradasi merata pada ketiga kelas akibat sparsity aktual yang jauh lebih tinggi (30,97%).



Gambar 5. Confusion Matrix Seluruh Varian Model TFLite Int8 pada Data Uji (N=721): *baseline*, Pruned 20%, 30%, 35%, 40%, 45%, 50%, dan 60%. Seluruh varian dievaluasi dengan prosedur dan data uji yang identik,

memperlihatkan bahwa degradasi klasifikasi signifikan hanya terjadi pada varian pruned50, yang memiliki sparsity aktual tertinggi (30,97%) di antara seluruh varian.

Tabel 3. *Classification Report* Model Proposed (40%+PTQ Int8, N=721)

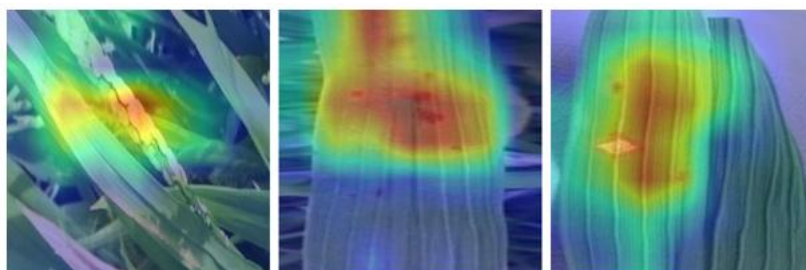
| Kelas                       | Precision | Recall | F1-Score | Support |
|-----------------------------|-----------|--------|----------|---------|
| BLB (Bacterial Leaf Blight) | 0,9647    | 0,9960 | 0,9802   | 247     |
| Brown Spot                  | 0,9920    | 0,9960 | 0,9940   | 249     |
| Leaf Blast                  | 0,9954    | 0,9556 | 0,9752   | 225     |
| Macro Average               | 0,9840    | 0,9825 | 0,9831   | 721     |

i Recall BLB=0.996 dan Brown Spot=0.996 dibulatkan menjadi 1.00 (2 desimal)

### 3.3 Analisis Interpretabilitas Grad-CAM

Perbandingan pada data uji yang konsisten (TFLite Int8 vs TFLite Int8) menunjukkan penurunan akurasi dari *baseline* 98,75% menjadi 98,34% pada model proposed, yaitu degradasi 0,41 poin persentase setara dengan sekitar 3 sampel tambahan yang salah klasifikasi dari 721 data uji. Penurunan ini tetap tergolong kecil mengingat reduksi ukuran model sebesar 88,23% dan mengonfirmasi bahwa PTQ Int8, bukan *pruning*, menjadi kontributor utama efisiensi ukuran pada konfigurasi ini (lihat catatan sparsity aktual pada Bagian 3.2). Penurunan akurasi ini disebabkan oleh dua faktor yang bekerja secara kumulatif. Pertama, kuantisasi Int8 mendiskritisasi bobot dan aktivasi dari representasi floating-point 32-bit menjadi rentang integer 8-bit, sehingga memunculkan galat pembulatan (*rounding error*) pada setiap operasi konvolusi; galat ini terakumulasi melalui lapisan-lapisan jaringan dan berpotensi menggeser batas keputusan (*decision boundary*) klasifikasi, terutama pada sampel yang secara visual berada di dekat batas antar kelas. Kedua, model pruned40 yang menjadi basis kuantisasi memiliki sparsity aktual yang rendah (2,29%, lihat Bagian 3.2) sehingga akurasi Keras-nya (99,45%) sudah sedikit lebih rendah dibandingkan *baseline* (99,58%) sebelum tahap PTQ diterapkan; PTQ kemudian menambahkan galat kuantisasi di atas selisih awal tersebut, sehingga total degradasi pada tahap TFLite (98,75% → 98,34%) merupakan efek gabungan dari kedua sumber galat, bukan semata-mata akibat kuantisasi. Sejalan dengan pola ini, analisis per kelas pada Bagian 3.3 menunjukkan bahwa penurunan recall paling nyata terjadi pada kelas Leaf Blast, yaitu kelas dengan ciri visual lesi awal yang belum sepenuhnya membentuk pola rhombus khas, sehingga representasi fiturnya lebih rentan terhadap galat diskretisasi dibandingkan kelas BLB dan Brown Spot yang memiliki pola visual lebih distingtif.

Analisis Grad-CAM dilakukan pada *layer 'out\_relu'* yang menghasilkan *feature map* 7×7 sebelum *Global Average Pooling*, konsisten dengan panduan Selvaraju et al. [12]. Hasil visualisasi (Gambar 6) menunjukkan: (1) BLB aktivasi tertinggi pada tepi daun, sesuai mekanisme infeksi melalui hidatoda/stomata margin daun [14]; (2) Brown Spot fokus pada bercak oval tunggal di tengah daun, mengenali pola konsentris eye-spot; (3) Leaf Blast aktivasi pada area berbentuk memanjang/diagonal yang merepresentasikan lesi rhombus *Magnaporthe oryzae*.



Gambar 6. Visualisasi Grad-CAM Model Proposed (40%+PTQ Int8): (a) BLB, (b) Brown Spot, (c) Leaf Blast

Validasi biologis ini mengkonfirmasi bahwa model mengekstrak fitur yang secara patologis relevan, bukan artefak statistik. Penggunaan Grad-CAM pada model terkompresi

(*pasca-pruning*) juga menunjukkan bahwa *pruning* 40% tidak merusak kapasitas representasi fitur biologis penting. Perlu dicatat bahwa interpretasi Grad-CAM dalam penelitian ini bersifat kualitatif-visual; analisis kuantitatif seperti pointing game accuracy atau intersection-over-union (IoU) dengan anotasi ground truth dari ahli patologi tanaman belum dilakukan dan menjadi rekomendasi untuk penelitian lanjutan guna memperkuat validitas interpretabilitas model.

### 3.4 Implikasi Deployment dan Keterbatasan

Model *proposed* (2,89 MB, 14,53 ms pada CPU Intel workstation) berpotensi kompatibel dengan *deployment* Android via TFLite Task Library, namun klaim kesiapan *deployment* ini bersifat prediktif dan belum divalidasi pada perangkat Android sesungguhnya.

Keterbatasan utama penelitian ini: (1) pengujian inferensi dilakukan pada CPU Intel workstation, bukan perangkat Android sesungguhnya validasi on-device direkomendasikan sebagai prioritas penelitian lanjutan; (2) dataset tidak mencakup kondisi lapangan nyata seperti pencahayaan tidak konsisten atau oklusi; (3) hanya tiga dari belasan penyakit padi Indonesia yang diuji. Studi terkait oleh Karim et al. [17] melaporkan latency ~20 ms pada edge device untuk model MobileNetV3 ukuran serupa, mengindikasikan potensi responsivitas real-time model ini jika divalidasi on-device. Hasil ini konsisten dengan temuan Khan et al. [18] yang menunjukkan kelayakan model berbasis MobileNet untuk *deployment* pada perangkat edge dengan keterbatasan sumber daya.

### 3.5 Implikasi Praktis untuk Petani Indonesia

Hasil penelitian ini memiliki implikasi praktis yang potensial bagi pertanian tanaman padi di Indonesia. Dengan akurasi deteksi 98,34% (TFLite Int8) pada model *proposed* dan reduksi ukuran model sebesar 88,23% melalui kombinasi *pruning* dan *Quantization*, sistem ini menunjukkan karakteristik yang sesuai untuk *deployment* pada perangkat mobile dengan keterbatasan daya komputasi, meskipun kelayakan praktisnya secara definitif baru dapat dipastikan setelah validasi on-device (lihat Bagian 3.5). Apabila tervalidasi, aplikasi berbasis model ini berpotensi dimanfaatkan petani Indonesia untuk: (1) deteksi dini penyakit daun padi di lapangan menggunakan smartphone Android secara offline; (2) pengurangan penggunaan pestisida yang tidak tepat sasaran melalui identifikasi penyakit yang lebih akurat; (3) percepatan pengambilan keputusan manajemen penyakit berbasis hasil deteksi yang tersedia secara real-time di lokasi.

Model yang dioptimalkan ini dirancang agar sesuai dengan spesifikasi perangkat kelas menengah-bawah (RAM 1–2 GB, storage 50–100 MB) yang umum ditemukan pada distribusi smartphone di wilayah pedesaan Indonesia; kesesuaian ini bersifat estimasi berdasarkan ukuran model (2,89 MB) dan belum diverifikasi melalui pengujian pada perangkat kelas tersebut secara langsung. Validasi lanjut bersama petani lokal serta pengujian kompatibilitas perangkat, sebagaimana direkomendasikan pada Bagian 3.5, menjadi prasyarat sebelum integrasi ke platform pertanian digital nasional pada fase implementasi berikutnya.

## 4. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengembangkan model deteksi penyakit daun padi terkompresi menggunakan kombinasi *magnitude-based weight pruning* dan *Post-Training Quantization* Int8 pada arsitektur MobileNetV2, dengan dua konfigurasi *sparsity* yang diuji: 40% dan 50%. Hasil eksperimen pada 721 data uji yang identik untuk seluruh konfigurasi menunjukkan bahwa model *proposed* (*pruning* 40% + PTQ Int8) mencapai akurasi 98,34% (F1-Score makro 0,9831) dengan ukuran 2,89 MB reduksi 88,23% dari *baseline .h5* (24,58 MB) dan *inference* time 14,53 ms/gambar pada CPU Intel workstation. Pemeriksaan lebih lanjut terhadap bobot model tersimpan menunjukkan bahwa *sparsity* aktual pada konfigurasi 40% hanya mencapai 2,29%, jauh di bawah target nominal, sehingga reduksi ukuran pada model ini terutama bersumber dari PTQ Int8, bukan dari pemangkasan bobot itu sendiri. Konfigurasi 50%, yang mencapai *sparsity* aktual mendekati target nominalnya (30,97%), menunjukkan degradasi akurasi yang berlangsung bertahap pada dua titik kompresi: akurasi turun dari baseline 99,58% menjadi 88,07% setelah

pruning (Keras, penurunan 11,51 poin persentase akibat pemangkasan bobot bermagnitudo tinggi yang sesungguhnya), kemudian turun lebih lanjut menjadi 83,50% setelah PTQ Int8 (penurunan tambahan 4,57 poin persentase akibat kuantisasi pada model yang bobotnya sudah terdegradasi), sehingga total degradasi kumulatif dari baseline hingga model 50%+PTQ Int8 mencapai 16,08 poin persentase. Pola bertahap ini menunjukkan bahwa sparsity 50% yang benar-benar tercapai merusak kualitas representasi model secara substansial sebelum kuantisasi diterapkan, dan PTQ Int8 selanjutnya memperberat degradasi tersebut. Berdasarkan perbandingan ini, 40% teridentifikasi sebagai konfigurasi dengan kinerja terbaik di antara dua tingkat *sparsity* yang diuji dalam penelitian ini; klaimnya dibatasi pada perbandingan kedua konfigurasi tersebut dan bukan sebagai ambang batas optimal secara umum, mengingat nilai *sparsity* lain seperti 20%, 30%, atau 45% belum diuji secara sistematis pada rentang *sparsity* aktual yang benar-benar tercapai. Analisis Grad-CAM mengonfirmasi bahwa fitur yang diekstraksi model pada *layer out\_relu* secara kualitatif relevan secara biologis untuk ketiga kelas penyakit, meskipun validasi kuantitatif interpretabilitas (misalnya *pointing game accuracy* atau IoU terhadap anotasi ahli) belum dilakukan.

Model terkompresi dalam format TFLite (2,89 MB) berpotensi diintegrasikan dengan aplikasi Android via TFLite Task Library dan beroperasi sepenuhnya *offline* karakteristik yang relevan untuk penggunaan di wilayah pertanian dengan keterbatasan koneksi internet. Namun, klaim kesiapan *deployment* ini bersifat prediktif: seluruh pengukuran *inference time* dalam penelitian ini dilakukan pada CPU workstation, bukan pada perangkat Android sesungguhnya, sehingga performa dan keandalan pada perangkat target belum dapat dipastikan. Penelitian lanjutan disarankan untuk: (1) validasi *on-device* pada perangkat Android berspesifikasi rendah (RAM 2 GB) guna mengonfirmasi *inference time* dan keandalan *deployment* nyata sebagai prasyarat sebelum model dapat dinyatakan siap digunakan di lapangan; (2) eksperimen sistematis pada tingkat sparsity tambahan (misalnya 20%, 30%, dan 45%) dengan penjadwalan *pruning* yang disesuaikan agar *sparsity* aktual benar-benar mencapai target nominal, guna memetakan hubungan *sparsity*–akurasi secara lebih komprehensif; (3) perluasan cakupan kelas penyakit padi Indonesia; (4) penerapan *Quantization-Aware Training* (QAT) [21] sebagai alternatif yang berpotensi mempertahankan akurasi lebih baik dibanding PTQ; dan (5) pengujian domain *generalization* pada dataset lapangan dari berbagai wilayah pertanian Indonesia dengan kondisi pencahayaan dan latar belakang yang bervariasi.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Dian Nuswantoro atas dukungan penelitian ini. Dataset diakses dari Mendeley Data (DOI: 10.17632/fwcj7stb8r/1) yang tersedia secara open access (CC BY 4.0).

## DAFTAR PUSTAKA

- [1] M. D. H. Bin Rahman and D. Zhang, "Trends in rice research: 2030 and beyond," *Food and Energy Security*, vol. 12, no. 4, e390, 2023, doi: 10.1002/fes3.390.
- [2] S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, "The global burden of pathogens and pests on major food crops," *Nature Ecology & Evolution*, vol. 3, no. 3, pp. 430–439, Feb. 2019, doi: 10.1038/s41559-018-0793-y.
- [3] B. Ribeiro et al., "Utilizing convolutional neural networks for the effective classification of rice leaf diseases through a deep learning approach," *Electronics*, vol. 13, no. 20, p. 4095, Oct. 2024, doi: 10.3390/electronics13204095.
- [4] G. Latif, S. E. Abdelhamid, R. E. Mallouhy, J. Alghazo, and Z. A. Kazimi, "Deep learning utilization in agriculture: Detection of rice plant diseases using an improved CNN model," *Plants*, vol. 11, no. 17, Sep. 2022, doi: 10.3390/plants11172230.
- [5] M. Muslih and E. H. Rachmawanto, "Convolutional neural network (CNN) untuk klasifikasi citra penyakit diabetes retinopathy," *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 5, no. 2, pp. 167–176, 2022, doi: 10.36080/skanika.v5i2.2945.

- [6] A. S. Kusuma, A. I. Pradana, and B. W. P., "Pengembangan sistem perhitungan jumlah kendaraan berdasarkan jenis kendaraan menggunakan algoritma YOLO secara realtime," *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 7, no. 2, pp. 166–179, 2024, doi: 10.36080/skanika.v7i2.3201.
- [7] S. Mulyana *et al.*, "Identifikasi penyakit tanaman berdasarkan citra daun berbasis web dengan pendekatan algoritma convolutional neural network," *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 8, no. 2, pp. 305–317, 2025, doi: 10.36080/skanika.v8i2.3573.
- [8] H. Guan, *et al.*, "A lightweight model for efficient identification of plant diseases and pests based on deep learning," *Frontiers in Plant Science*, vol. 14, p. 1227011, 2023, doi: 10.3389/fpls.2023.1227011.
- [9] M. Sandler, *et al.*, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
- [10] H. Nugroho, J. X. Chew, S. Eswaran, and F. S. Tay, "Resource-optimized CNNs for real-time rice disease detection with ARM Cortex-M microprocessors," *Plant Methods*, vol. 20, no. 159, 2024, doi: 10.1186/s13007-024-01280-6.
- [11] O. O. Martins, C. C. Oosthuizen, and D. A. Desai, "RiceLeafClassifier-v1.0: A quantized deep learning model for automated rice leaf disease detection and edge deployment," *Engineering Reports*, vol. 7, no. 6, p. e70231, 2025, doi: 10.1002/eng2.70231.
- [12] 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. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.
- [13] H. B. Prajapati, J. P. Shah, and V. K. Dabhi, "Detection and classification of rice plant diseases," *Intelligent Decision Technologies*, vol. 11, no. 3, pp. 357–373, 2017, doi: 10.3233/IDT-170301.
- [14] B. S. Bari *et al.*, "A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework," *PeerJ Computer Science*, vol. 7, p. e432, 2021, doi: 10.7717/peerj-cs.432.
- [15] A. S. Paymode and V. B. Malode, "Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG," *Artificial Intelligence in Agriculture*, vol. 6, pp. 23–33, 2022, doi: 10.1016/j.aiia.2021.12.002.
- [16] A. Haridasan, J. Thomas, and E. D. Raj, "Deep learning system for paddy plant disease detection and classification," *Environmental Monitoring and Assessment*, vol. 195, no. 120, 2023, doi: 10.1007/s10661-022-10656-x.
- [17] M. J. Karim, *et al.*, "Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight CNN architecture and Grad-CAM," *Scientific Reports*, vol. 14, no. 16022, 2024, doi: 10.1038/s41598-024-66989-9.
- [18] A. T. Khan, S. M. Jensen, A. R. Khan, and S. Li, "Plant disease detection model for edge computing devices," *Frontiers in Plant Science*, vol. 14, p. 1308528, 2023, doi: 10.3389/fpls.2023.1308528.
- [19] B. N. Kumar and S. Sakthivel, "Rice leaf disease classification using a fusion vision approach," *Scientific Reports*, vol. 15, no. 8692, 2025, doi: 10.1038/s41598-025-87800-3.
- [20] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Advances in Neural Information Processing Systems*, vol. 28, 2015, pp. 1135–1143.
- [21] B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2704–2713, doi: 10.1109/CVPR.2018.00286.
- [22] P. K. Sethy, "Rice leaf disease image samples," *Mendeley Data*, vol. 1, 2020, doi: 10.17632/fwcj7stb8r.1.
- [23] International Rice Research Institute (IRRI), "Bacterial blight," *Rice Knowledge Bank*. [Online]. Available: <http://www.knowledgebank.irri.org/decision-tools/rice-doctor/rice-doctor-fact-sheets/item/bacterial-blight>.

- [24] M. Bilal et al., "High-performance deep learning for instant pest and disease detection in precision agriculture," *Food Science & Nutrition*, 2025, doi: 10.1002/fsn3.70963.
- [25] H. K. Kondaveeti and C. G. Simhadri, "Evaluation of deep learning models using explainable AI with qualitative and quantitative analysis for rice leaf disease detection," *Scientific Reports*, vol. 15, no. 31850, 2025, doi: 10.1038/s41598-025-14306-3.

---

i Recall BLB=0.996 dan Brown Spot=0.996 dibulatkan menjadi 1.00 (2 desimal)