

## BAB V PENUTUP

### 5.1. Kesimpulan

Kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan mengenai perancangan aplikasi pendeteksi penyakit padi menggunakan algoritma EfficientNetV2-S berbasis *Android* adalah sebagai berikut:

1. Pembangunan Model *Deep Learning* EfficientNetV2-S

Model EfficientNetV2-S berhasil dibangun menggunakan pendekatan *transfer learning* yang terdiri dari tahap *feature extraction* dan *fine-tuning* untuk menyesuaikan model dengan karakteristik citra daun padi. Berdasarkan hasil evaluasi pada lingkungan pengembangan seperti *Kaggle Notebook* dan *Google Collaboratory*, model mampu mencapai tingkat akurasi sebesar 89% dengan nilai *Precision* dan *F1-score* yang cukup baik pada sebagian besar kelas penyakit. Hal ini menunjukkan bahwa model memiliki kemampuan yang baik dalam mengenali pola visual pada citra daun padi. Namun demikian, ketika model diimplementasikan dan diuji pada aplikasi *Android*, performa deteksi yang dihasilkan cenderung menurun. Penurunan ini dipengaruhi oleh beberapa faktor, antara lain perbedaan lingkungan eksekusi, proses konversi model ke format *TensorFlow Lite* (TFLite) yang melibatkan kompresi, serta ketidakteraturan teknik pengambilan gambar pada dataset, seperti variasi sudut pengambilan, pencahayaan, dan kualitas kamera. Faktor-faktor tersebut menyebabkan model tidak selalu menghasilkan deteksi yang konsisten pada perangkat ponsel pintar.

2. Integrasi Model EfficientNetV2-S ke Aplikasi *Android*

Model yang telah dilatih dikonversi ke dalam format *TensorFlow Lite* (.tflite) agar dapat dijalankan pada perangkat *Android* dengan keterbatasan sumber daya. Model dan label kelas penyakit kemudian diintegrasikan ke dalam proyek *Android Studio* dan dikembangkan menggunakan bahasa pemrograman Kotlin dengan arsitektur *Model-View-ViewModel* (MVVM). Aplikasi mampu melakukan proses inferensi dan menampilkan hasil deteksi penyakit padi secara *real-time*. Meskipun demikian, hasil integrasi menunjukkan bahwa performa model pada aplikasi *Android* masih belum optimal. Hal ini ditandai dengan adanya penurunan akurasi deteksi dibandingkan dengan hasil pengujian pada lingkungan pengembangan. Kondisi tersebut menunjukkan bahwa diperlukan penyesuaian lebih lanjut, khususnya pada tahap praproses citra, konfigurasi inferensi TFLite, serta optimasi model agar performa pada perangkat *Android* dapat lebih mendekati performa model asli.

### 3. Pengujian Aplikasi dengan *Black-box Testing*

Pengujian aplikasi dilakukan menggunakan metode *black-box testing* untuk mengevaluasi fungsi-fungsi utama aplikasi dari sudut pandang pengguna. Hasil pengujian menunjukkan bahwa sebagian besar fitur aplikasi dapat berjalan sesuai dengan kebutuhan fungsional yang dirancang, seperti pemilihan gambar, proses deteksi, serta penampilan informasi penyakit padi. Aplikasi dapat dijalankan pada perangkat *Android* dengan versi minimal 7.0 “Nougat” (SDK 24) dan memiliki ukuran instalasi sebesar 110 MB. Namun, hasil pengujian juga menunjukkan bahwa aplikasi belum sepenuhnya memberikan hasil deteksi yang akurat pada seluruh skenario pengujian. Keterbatasan ini berkaitan erat dengan performa model yang menurun setelah dikonversi ke format TFLite dan diintegrasikan ke dalam aplikasi *Android*. Oleh karena itu, meskipun aplikasi telah berfungsi secara fungsional, masih diperlukan pengembangan dan penyempurnaan lebih lanjut agar sistem dapat memberikan hasil deteksi yang lebih andal dan konsisten bagi pengguna akhir.

## 5.2. Saran

Berdasarkan hasil penelitian yang telah dicapai, berikut beberapa saran untuk pengembangan penelitian dan aplikasi di masa mendatang:

### 1. Konsistensi Teknik Pengambilan Dataset

Disarankan agar dataset yang digunakan memiliki teknik pengambilan gambar yang konsisten, meliputi latar belakang, posisi daun, jarak pengambilan, serta tingkat pencahayaan. Konsistensi ini penting agar model dapat lebih mudah mempelajari pola visual berupa bercak atau gejala penyakit pada daun padi, sehingga akurasi deteksi dapat ditingkatkan dan lebih stabil saat digunakan di lapangan.

### 2. Keseimbangan Jumlah Data pada Setiap Kelas

Untuk menghindari bias model terhadap kelas tertentu, disarankan agar setiap kelas penyakit memiliki jumlah data yang relatif seimbang. Ketidakseimbangan data dapat menyebabkan model lebih dominan mendeteksi kelas mayoritas dan menurunkan performa pada kelas minoritas. Pada penelitian selanjutnya, ketidakseimbangan data dapat diatasi dengan menerapkan *Synthetic Minority Oversampling Technique* (SMOTE), sebagaimana digunakan pada penelitian terdahulu yang ditunjukkan pada Tabel 2.3, guna menghasilkan data sintesis pada kelas minoritas sehingga distribusi data menjadi lebih proporsional.

### 3. Penyesuaian Tahap Praproses Citra

Setiap arsitektur model *Deep Learning* memiliki standar praproses citra yang berbeda. Oleh karena itu, disarankan untuk memastikan bahwa proses praproses, seperti normalisasi, ukuran citra, dan skema warna, disesuaikan dengan kebutuhan model yang digunakan. Praproses yang tepat akan membantu model dalam mengekstraksi fitur secara optimal dan meningkatkan performa deteksi.

4. Keselarasan Konfigurasi Model TFLite dan Inferensi *Android*

Dalam proses konversi model ke format *TensorFlow Lite*, diperlukan perhatian khusus agar konfigurasi model tetap konsisten dengan model asli yang dilatih. Hal ini mencakup kesesuaian format input, skema normalisasi, ukuran citra, serta pengaturan inferensi pada *Android Studio*. Ketidaksesuaian pada tahap ini dapat menyebabkan penurunan performa model ketika dijalankan di perangkat *Android*, meskipun model menunjukkan hasil yang baik pada lingkungan pengembangan.