

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil analisis dan implementasi yang telah dilakukan, empat model arsitektur *Convolutional Neural Network*, yaitu ResNet50, DenseNet169, Inception V3, dan MobileNet V3, dibandingkan untuk klasifikasi citra sampah yang terdiri dari enam kelas: kaca, kardus, kertas, logam, organik, dan plastik. Hasil skripsi dijelaskan sebagai berikut:

1. Berdasarkan evaluasi model ResNet50, DenseNet169, Inception V3, dan MobileNet V3 untuk klasifikasi citra sampah dalam enam kelas, dapat disimpulkan bahwa DenseNet169 menunjukkan performa terbaik dengan mencapai akurasi 93,33% pada pengujian menggunakan *confusion metrics*. DenseNet169 juga memiliki nilai *precision* dan *recall* yang sangat tinggi di hampir semua kelas, serta *F1-Score* yang konsisten tinggi. ResNet50 juga menunjukkan kinerja yang baik dengan akurasi 89,09%, serta *precision* dan *recall* yang tinggi dan seimbang, menjadikannya model terbaik setelah DenseNet169. Inception V3 memiliki performa yang cukup baik dengan akurasi 84,45%, meskipun sedikit kurang optimal di beberapa kelas berdasarkan hasil evaluasinya. MobileNet V3 menunjukkan performa terlemah dengan hanya mencapai akurasi 35,15%, serta *precision* dan *recall* yang rendah, terutama pada kelas *Glass*, *Metal*, dan *Organic*.

Oleh karena itu, DenseNet169 adalah model terbaik untuk tugas ini, diikuti oleh ResNet50, sementara MobileNet V3 tidak direkomendasikan.

2. Pembuatan aplikasi Android menggunakan bahasa pemrograman Kotlin telah berhasil menyediakan antarmuka untuk melihat hasil prediksi langsung dari model dengan akurasi terbaik, yaitu DenseNet169. Aplikasi ini dikembangkan dengan bantuan library TensorFlow *Lite* dan menggunakan model yang telah dikonversi menjadi format .tflite. Hasilnya, aplikasi dapat memprediksi jenis citra sampah yang diunggah oleh pengguna.

## 5.2 Saran

Adapun saran yang dapat diberikan untuk pengembangan skripsi ini di masa mendatang adalah sebagai berikut:

1. Menambahkan jumlah jenis kelas sampah seperti kain, untuk dapat mengklasifikasikan lebih banyak jenis sampah. Jenis sampah lain yang juga dapat dipertimbangkan termasuk elektronik, baterai, dan sampah berbahaya lainnya. Hal ini akan memperluas kemampuan sistem dalam mengidentifikasi berbagai kategori sampah yang lebih beragam dan spesifik, sehingga dapat mendukung program daur ulang yang lebih komprehensif.
2. Menambahkan jumlah dataset dan variasi yang lebih beragam supaya dapat mendeteksi jenis sampah lebih baik lagi. Ini berarti mengumpulkan lebih banyak gambar untuk setiap kategori sampah yang ada, termasuk kardus, kertas, logam, kaca, organik, dan plastik. Selain itu, variasi yang lebih beragam meliputi pengumpulan gambar dengan berbagai kondisi pencahayaan, sudut pengambilan

gambar, dan latar belakang yang berbeda-beda. Hal ini akan membantu model menjadi lebih *robust* dan akurat dalam mengklasifikasikan sampah di berbagai situasi nyata, meningkatkan kemampuan model untuk mengenali jenis sampah meskipun dalam kondisi yang berbeda dari *dataset* pelatihan.

3. Menerapkan teknis pra-pemrosesan gambar yang berbeda untuk mengurangi *noise* dari latar belakang citra gambar yang dimiliki.
4. Mengeksplorasi model-model arsitektur lain yang mungkin menawarkan kinerja yang lebih baik atau lebih efisien. Selain model-model yang telah digunakan, ada beberapa model lain yang layak dipertimbangkan. Misalnya, EfficientNet, yang dikenal karena efisiensinya dan performanya yang baik dalam berbagai tugas klasifikasi, atau *Vision Transformers* (ViT). Model lain seperti NasNet dan Xception juga dapat dieksplorasi untuk melihat apakah model tersebut dapat menawarkan keunggulan dalam hal akurasi atau kecepatan inferensi.
5. Mengeksplorasi metode *deploy* dan integrasi *Machine Learning* ke aplikasi atau web yang lain. Karena TFLite memiliki kekurangan, yaitu dapat menyebabkan ukuran aplikasi membengkak, metode lain perlu dipertimbangkan. Selain menggunakan TFLite, terdapat beberapa metode lain yang dapat dipertimbangkan. Misalnya, layanan cloud seperti Google Cloud AI, AWS SageMaker, atau Azure ML untuk *deployment*, terutama untuk aplikasi yang memerlukan skala dan ketersediaan tinggi. Integrasi model ML ke aplikasi web dapat dilakukan dengan menggunakan API RESTful yang dihosting di *server cloud*. Untuk aplikasi *mobile*, selain TFLite, model juga dapat diintegrasikan menggunakan Core ML untuk iOS atau NNAPI untuk Android.

6. Mengembangkan kemampuan model untuk mendeteksi dan mengklasifikasikan lebih dari satu jenis sampah dalam satu gambar dengan menggunakan teknik *object detection*, seperti menggunakan arsitektur seperti YOLO (*You Only Look Once*) atau SSD (Single Shot MultiBox Detector). Teknik ini memungkinkan model untuk mendeteksi dan mengklasifikasikan beberapa objek dalam satu gambar, sehingga meningkatkan kegunaan dan fleksibilitas sistem dalam berbagai situasi nyata.
7. Menambahkan rekomendasi pendauran ulang pada aplikasi. Selain mengklasifikasikan jenis sampah, aplikasi dapat dilengkapi dengan fitur yang memberikan rekomendasi tentang cara mendaur ulang setiap jenis sampah yang terdeteksi. Misalnya, aplikasi dapat memberikan informasi tentang pusat daur ulang terdekat, jenis material yang dapat didaur ulang, dan tips untuk mempersiapkan sampah sebelum didaur ulang. Fitur ini akan membantu meningkatkan kesadaran dan partisipasi masyarakat dalam program daur ulang, serta mendukung upaya pengelolaan sampah yang lebih efektif dan berkelanjutan.