

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan analisa data dari penelitian, peneliti dapat mengambil kesimpulan sebagai berikut :

1. Hasil implementasi dan evaluasi menunjukkan bahwa model *deep learning* dengan metode *transfer learning* pada model MobileNetV3 dapat mengklasifikasikan kondisi jalan dengan sangat baik. Dari keenam model yang dilatih, peneliti memilih model MobileNetV3 varian *large* dengan rasio *dataset* 60:20:20. Hasil evaluasi model MobileNetV3 varian *large*, menunjukkan nilai *accuracy* 0,9004; *precision* 0,8759; *recall* 0,9330 dan *F1-Score* 0.9035. Setelah dikonversi menjadi model Tensorflow Lite, model memiliki nilai *accuracy* 0,8935; *precision* 0,8585; *recall* 0,9423 dan *F1-Score* 0,8985.
2. Hasil pengembangan aplikasi android dalam mengklasifikasikan kondisi jalan dapat berfungsi dengan baik dan memberikan peringatan dini pada pengendara berhasil diimplementasikan. Pada menu *Find*, pengguna dapat memasukkan tujuan dan aplikasi akan menampilkan rute serta titik berlubang yang pada rute tersebut. Apabila ada lubang dalam jarak 100m terhadap pengguna, aplikasi akan memberikan peringatan baik secara visual maupun secara audio. Pada menu data, pengguna dapat melihat daftar jalan yang rusak ataupun normal. Selain itu, pada menu kontribusi, pengguna dapat mengklasifikasikan kondisi jalan dan dapat membantu pengguna lain dengan menyinkronkan riwayat hasil klasifikasi kondisi jalan. Kedua menu tersebut membantu pengguna dalam hal mengantisipasi kecelakaan dan permasalahan dalam perjalanan yang disebabkan karna adanya jalan yang berlubang.

5.2 Saran

Berikut ini adalah beberapa saran dari penulis untuk penelitian selanjutnya, yaitu:

1. Penelitian selanjutnya dapat mencoba menggunakan ukuran gambar yang lebih besar, misalnya gambar dengan ukuran 448x448 atau 512x512.
2. Menggunakan kombinasi augmentasi data yang berbeda untuk meningkatkan akurasi model.
3. Menggunakan *pre-trained model* yang berbeda, misalnya EfficientNetV2.
4. Menambahkan jumlah *dataset* yang seimbang pada kelas berlubang dan normal sehingga performa dari model dapat menjadi lebih baik.
5. Menggunakan teknik optimisasi model TensorFlow Lite yang berbeda, misalnya dengan menggunakan *quantization* untuk menurunkan *inference time*.
6. Memperhitungkan lajur posisi pengendara, ukuran lubang dan posisi lubang sehingga peringatan yang diberikan lebih berguna pada pengguna.