

BAB 5. KESIMPULAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat diambil kesimpulan

1. Model *YOLOv8* dapat diterapkan melalui pengumpulan dataset guiding block, yang kemudian dibagi ke dalam data pelatihan, validasi, dan pengujian menggunakan platform *Roboflow*. Proses pelatihan dilakukan menggunakan model *YOLOv8n*, yang dipilih karena ringan dan cocok untuk diimplementasikan ke dalam perangkat mobile. Model dilatih selama 100 epoch dan menghasilkan performa terbaik dengan nilai mAP sebesar 0.959 beserta mAP untuk setiap kelasnya seperti kelas lurus sebesar 0.979, kelas kanan sebesar 0.968, dan kelas kiri sebesar 0.932. Pemilihan *YOLOv8n* juga mempertimbangkan efisiensi waktu pelatihan, yang hanya membutuhkan sekitar 31 detik per epoch.
2. Model yang telah dibangun kemudian diimplementasikan ke dalam aplikasi *mobile* menggunakan *TensorFlow Lite* (TFLite) dan dikembangkan agar dapat mendeteksi guiding block secara real-time melalui kamera. Aplikasi ini juga ditambahkan dengan fitur *Text-to-Speech* (TTS) untuk membacakan hasil prediksi kepada pengguna sehingga dapat memberikan bantuan navigasi arah *guiding block* bagi penyandang tunanetra. Proses pengujian dilakukan dengan metode *blackbox* yang mencakup berbagai kondisi cahaya seperti siang dan malam hari beserta pengujian jika terdapat objek *guiding block* dan tidak. Seluruh pengujian berhasil dilakukan dan menunjukkan hasil VALID yang membuktikan bahwa aplikasi berfungsi dengan baik dan stabil dalam berbagai situasi. Aplikasi ini juga memiliki kecepatan deteksi objek yang menyentuh 13 hingga 18 FPS menggunakan model *YOLOv8n*.

5.2. Saran

Untuk pengembangan lebih lanjut, disarankan agar aplikasi ini dapat lebih dikembangkan dengan menambahkan variasi objek lainnya yang lebih kompleks dan relevan dengan kebutuhan pengguna tunanetra, seperti pertigaan, perempatan, tanda berhenti, atau simbol lain yang umum ditemui di jalur pejalan kaki. Penambahan kelas ini bertujuan agar arah yang diberikan oleh sistem tidak hanya terbatas pada jalan lurus, belok kiri, dan belok kanan, tetapi juga dapat memberi arahan yang lebih menyesuaikan kondisi lingkungan sekitar.

Selain itu, disarankan juga untuk meningkatkan tingkat akurasi karena terdapat kekurangan dalam mendeteksi kelas kanan yang menyebabkan sering kali menjadi deteksi kelas kiri beserta kekurangan dalam memberikan informasi navigasi sehingga dapat ditambahkan perhitungan jarak (*distance equation*) antara pengguna dan objek *guiding block* yang terdeteksi. Fungsi ini penting untuk

mengetahui seberapa jauh objek berada dari posisi pengguna dan dapat memberikan estimasi waktu atau peringatan yang lebih dini.