

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berikut merupakan kesimpulan yang ditarik dari penelitian ini adalah sebagai berikut:

1. Sebuah sistem jaringan deteksi kendaraan berbasis algoritma YOLO telah dibuat dengan pembagian kelas sebanyak 4 kelas diantaranya *bus* TransJakarta, Sepeda Motor, Mobil, dan Truk. Proses pembuatan sistem dilakukan menggunakan Google colab dan API Roboflow yang saling terhubung dengan total *dataset* sebanyak 533 citra. Skenario pengujian pada penelitian ini diantaranya pengujian di kondisi lalu lintas yang sepi, kondisi lalu lintas yang ramai, dan lokasi yang berbeda dari *dataset* dengan berbagai sudut pandang untuk menganalisa kinerja model YOLO.
2. Dilakukan pengujian pada skenario diatas, nilai performa jaringan yang didapat tercapai dengan nilai yang baik diantaranya nilai mAP 92,2%, nilai *precision* 89,5% dan nilai *recall* 84,2% dari total 300 epochs. Dapat disimpulkan bahwa sistem deteksi objek penelitian kali ini dapat bekerja dengan sangat baik. Kemudian terdapat perbedaan hasil yang cukup signifikan dari *confidence score* terhadap 3 skenario pengujian diatas yang dipengaruhi oleh *angle distortion* sudut pandang, tumpang tindih objek kendaraan yang disebabkan kendaraan saling berdekatan dalam satu jalur lalu lintas. Dengan kondisi terbaik ada pada kondisi lalu lintas normal dan sudut pandang yang konsisten terhadap *dataset* yang sudah dilatih.

#### **5.2 Saran**

Untuk pengembangan berdasarkan hasil penelitian ini, penulis memberikan beberapa rekomendasi untuk mendukung pengembangan penelitian selanjutnya, antara lain :

1. Menghubungkan 2 teknologi terkini yaitu pengolahan citra dan mikrokontroler untuk dapat diimplementasikan langsung di lapangan. Salah satu teknologi yang dapat diterapkan adalah gerbang otomatis yang bekerja berdasarkan hasil pengolahan citra untuk menyeleksi kendaraan yang berhak melintas di jalur TransJakarta.
2. Menambah variasi *dataset* dengan berbagai kondisi untuk meningkatkan performa model dalam deteksi objek serta mempertimbangkan penggunaan arsitektur lain seperti *faster R-CNN*, *SSD*, *RetinaNet* dan lain – lain untuk memperluas variasi dan pemahaman tentang performa model yang berbeda.