

Implementasi *Framework Mask R-CNN Object Detection API* Dalam Mengklasifikasi Jenis Kendaraan Bermotor

Dadang Iskandar Mulyana, Apian Candra Aditya, Seli Amelia, Sony Agustiansyah

Teknik Informatika, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika
Jl. Raden Inten II No.8, Duren Sawit, Kota Jakarta Timur, Indonesia
Email: ameliaseli092@gmail.com

Naskah Masuk: 05 Februari 2022; Diterima: 23 Februari 2022; Terbit: 25 Maret 2022

ABSTRAK

Abstrak – Jenis jenis kendaraan yang melewati suatu lalu lintas pada jalan sangat berpengaruh terhadap kepadatan lalu lintas pada sebuah jalan. Jadi bila jumlah setiap jenis kendaraan yang melewati suatu lalu lintas pada jalan dapat diketahui, maka taktik lalu lintas yang baik dapat diatur oleh sang pihak yang berwenang didaerah tersebut. Melalui pemanfaatan perkembangan teknologi di dalam bidang *computer vision* serta *deep learning*, penelitian ini bertujuan untuk mendeteksi dan serta menghitung objek kendaraan yang melewati suatu lalu lintas pada jalan sesuai dengan pembagian terstruktur mengenai jenis jenis kendaraan. Penelitian ini memakai *Framework Mask R-CNN Object Detection API* dan algoritma dari *pretrained* contoh *object detection YOLO v3* untuk melakukan deteksi dan penjabaran. Berdasarkan yang akan terjadi pengujian pada kelima video uji coba tersebut, dengan sistem yang dibangun berhasil membentuk suatu data yang berisi jumlah setiap jenis jenis tunggangan yang lewat di suatu lintas pada jalan dengan tingkat akurasi deteksi objek dengan rata-rata adalah 90.8%.

Kata kunci: *Deep Learning, Object Detection, Mask R-CNN, YOLO, Kendaraan Bermotor*

ABSTRACT

Abstract - The types of vehicles that pass through a traffic on the road greatly affect the density of traffic on a road so if the number of each type of vehicle that passes through a traffic on the road can be known, then good traffic tactics can be arranged by the authorities in the area. . utilizing technological developments in the field of computer vision and deep learning, this study aims to detect and calculate vehicle objects that pass through a traffic on the road according to a structured division of vehicle types. This research uses the *R-CNN Object Detection API Mask Framework* and the algorithm from the *pretrained YOLO v3 object detection sample* to detect and describe. based on what will happen in the testing of the five video trials, the system that was built succeeded in forming a data containing the number of each type of ride that passed in a traffic on the road with an average object detection accuracy rate of 90.8%.

Keywords: *Deep Learning, Object Detection, Mask R-CNN, YOLO, Motor vehicle*

Copyright © 2022 Universitas Muhammadiyah Jember.

1. PENDAHULUAN

Lancarnya suatu lalu lintas dipengaruhi oleh beberapa faktor, antara lain luas jalan, jumlah kendaraan dan jenis kendaraan yang lewat. Seiring berjalannya waktu, lalu lintas semakin dipenuhi dengan berbagai jenis kendaraan. Ditambah dengan luas jalan yang sudah tidak dapat menampung jumlah kendaraan yang berlalu lintas dan strategi lalu lintas yang kurang baik membuat arus lalu lintas suatu jalan semakin padat dan kemacetan semakin berbondong-bondong. Hal ini sangat berpengaruh besar di dalam kehidupan sehari-hari manusia, misalkan waktu yang terbuang cukup banyak.

Salah satu hal yang dapat dilakukan untuk mengurangi kemacetan adalah mengatur strategi lalu lintas yang baik. Jika terdapat data jenis kendaraan yang lewat di suatu jalan selama waktu tertentu, maka pihak yang berwenang dapat menganalisa dan mengambil keputusan untuk mengatur strategi lalu lintas yang baik. Misalkan jika di suatu jalan yang macet ternyata jumlah kendaraan mobil berjenis truk yang lewat memiliki kuantitas yang lebih banyak dibandingkan jenis kendaraan yang lain, maka pihak yang berwenang dapat memberikan jalur sendiri terhadap kendaraan mobil berjenis truk. Oleh karena itu, untuk mengatur

taktik lalu lintas yang baik, maka pihak yang berwenang membutuhkan data yang berisi jumlah kendaraan yang lewat di suatu jalan berdasarkan jenis kendaraan. Pekerjaan menghitung jumlah kendaraan berdasarkan jenis kendaraan yang lewat di suatu jalan setiap hari merupakan pekerjaan yang tidak mudah bagi manusia, karena harus membutuhkan kesabaran dan ketelitian.

Dalam perkembangan zaman dan teknologi, *deep learning* semakin berkembang dengan sangat pesat di dalam menciptakan program-*application synthetic intelligence*. Deep learning menggunakan *artificial neural network* yang membuat *deep learning* mampu untuk terus melatih diri sendiri dalam mengenali pola yang benar berdasarkan facts atau input yang diberikan. Mulai dari *face detection*, *face recognition*, *item detection* dan *object recognition* mulai diciptakan menggunakan *deep learning*. Seiring dengan berjalannya waktu, *framework-framework deep learning* pun mulai dibuat. Mask R-CNN merupakan salah satu *framework deep learning* yang dapat digunakan untuk membuat berbagai program *artificial intelligence*. Oleh karena itu, dengan memanfaatkan *framework deep learning*, yaitu Mask R-CNN, maka dapat memungkinkan untuk membuat sebuah sistem deteksi objek dalam menghitung jumlah kendaraan yang lewat di suatu jalan berdasarkan klasifikasi jenis kendaraan

Dari latar belakang penelitian ini, maka dibangunlah sebuah sistem atau aplikasi untuk mendeteksi dan menghitung jumlah kendaraan yang lewat di suatu jalan berdasarkan klasifikasi jenis kendaraan menggunakan framework Mask R-CNN object detection. Klasifikasi jenis kendaraan yang akan dideteksi antara lain, jenis kendaraan mobil, truk, bus dan sepeda motor. Data yang dihasilkan berupa jumlah kendaraan yang lewat di suatu jalan berdasarkan klasifikasi jenis kendaraan, di mana data tersebut dapat digunakan oleh pihak yang berwenang untuk mengatur strategi lalu lintas yang baik. Adapun topik yang diangkat pada penelitian ini adalah "Implementasi Framework Mask R-CNN Object Detection API Dalam Mengklasifikasi Jenis Kendaraan Bermotor".

2. KAJIAN PUSTAKA

2.1 Penelitian Terkait

Terdapat beberapa penelitian yang terkait dengan penelitian saat ini mengenai implementasi deteksi objek untuk mendeteksi dan mengklasifikasi objek kendaraan, serta menghitung jumlah objek kendaraan yang melewati suatu jalan. Pada penelitian sebelumnya yang berjudul "Analisis Pelacakan Objek Menggunakan *Background Estimation* Pada Kamera Diam Dan Bergerak" menggunakan metode *background estimation*. Hasil dari penelitian ini adalah uji coba dilakukan terhadap objek video kendaraan beroda empat selanjutnya dilakukan analisis dan diperoleh hasil akhir berupa identifikasi serta pemisahan antara objek kendaraan beroda empat dengan background-nya. Objek yang dipergunakan pada pengujian merupakan objek video mobil yg diambil dengan memakai kamera diam (tidak aktif) serta kamera beranjak. hasil pengujian analisis pelacakan mobil yg dilakukan terhadap nilai parameter menggunakan memakai kamera diam (tidak aktif) berhasil menangkap objek mobil dan memisahkan backround asal objeknya. Sedangkan pengujian analisis pelacakan mobil terhadap nilai parameter dengan memakai kamera beranjak tidak berhasil mengidentifikasi objek kendaraan beroda empat dan memisahkan background, sebab semua objek kendaraan beroda empat yg melintas dan background dari objek ikut terdeteksi sebagai satu kesatuan [1].

Penelitian lain yang berjudul "Analisis Perbandingan Pelacakan Objek memakai *Optical Flow* serta *Background Estimation* di Kamera bergerak" menggunakan metode *optical flow* dan *background estimation*. Hasil dari penelitian ini ialah Pengujian *optical flow* berhasil mengidentifikasi objek yang melintas. Pengujian *background estimation* tidak berhasil mengidentifikasi objek serta tidak dapat membedakan antara objek dengan background. Pengujian dipandang dari jarak objek menggunakan kamera saat direkam, banyak dan sedikitnya *background* dan kecepatan yang ditempuh sang objek ketika diteliti [2].

Penelitian lain yang berjudul "*Deep Learning RetinaNet Based Car Detection for Smart Transportation Network*" menggunakan metode STN yaitu *Infrastructure-to-Vehicle (I2V)*. Hasil dari penelitian ini adalah hasil deteksi mentrigger VLC transmitter yang terpasang pada lampu penerangan jalan mengirimkan sinyal info menuju VLC receiver yang dipasang pada kendaraan beroda empat. Pada tahap proses pembinaan, jumlah dataset mobil yang digunakan merupakan sekitar 1600 image serta 400 validation image dan pengulangan proses sebesar 100 epoch. berdasarkan 50 kali pengujian pada image test, diperoleh nilai precision mencapai 86%, nilai recall mencapai 85% dan f1-score mencapai 84% [3].

Penelitian lain yang berjudul "Deteksi Halangan Menggunakan Metode Stereo R-CNN pada mobil Otonom" menggunakan metode Algoritma Stereo *Regional Convolutional Neural Network* (Stereo R-CNN). Hasil dari penelitian ini adalah metode ini dapat mendeteksi bounding box 3D dari objek pada gambar. Algoritma ini dapat mendeteksi objek dengan jarak deteksi yang lebih jauh dibandingkan dengan LIDAR, namun saat ini metode ini masih kurang feasible untuk dapat diterapkan sebagai pengganti LIDAR

sebagai sensor utama karena kecepatan deteksinya yang masih lambat di kisaran 0.81 detik (1,2 frame per detik) [4].

Penelitian lain yang berjudul “Deteksi Mata Mengantuk Pada Pengemudi Mobil Menggunakan Metode Viola Jones” menggunakan metode Viola Jones. Hasil dari penelitian ini adalah menghasilkan output berupa suara peringatan bagi pengendara kendaraan mobil yang mengantuk atau tidak mengantuk secara otomatis. Pada pengujian program ini didapatkan jumlah yang terdeteksi 7 mata dari 10 mata dengan memakai level BW 0.255 yang berguna untuk mempercepat suatu program untuk mendeteksi mata yang mengantuk [5].

2.2 Convolutional Neural Networks

Artificial neural networks (ANNs) atau biasa disebut dengan jaringan saraf tiruan adalah sebuah sistem yang digunakan untuk proses komputasi yang terinspirasi dari sistem saraf biologis, seperti otak manusia. *Artificial neural networks* (ANNs) terdiri dari sejumlah besar node komputasi yang saling berhubungan (neuron) yang secara kolektif belajar dari input untuk mengoptimalkan hasil akhir. Cara kerja *convolutional neural networks* (CNNs) sama dengan tradisional *artificial neural networks* di mana keduanya terdiri dari sekumpulan neuron yang mengoptimalkan diri dari pembelajaran. Mulai dari input awal sampai pada hasil akhir, setiap jaringan akan tetap mengekspresikan sebuah skor atau yang disebut dengan weight dan pada layer terakhir terdapat fungsi kehilangan (*loss functions*) yang terkait dengan kelas yang ada.

2.3 Deep Learning

Teknologi *Machine Learning* telah sangat banyak membantu dalam perkembangan dan kemajuan berbagai aspek komunitas modern saat ini, seperti pencarian web atau pun penyaringan konten dalam sosial media. *Machine learning* juga membantu dalam pengembangan produk-produk saat ini, seperti kamera, smartphone dan lain sebagainya. Sistem *machine learning* digunakan untuk mengidentifikasi objek dalam gambar, mengubah suara ke dalam bentuk teks, menampilkan hasil yang relevan terhadap pencarian yang dilakukan dan masih banyak lagi. Seiring dengan berjalannya waktu, aplikasi-aplikasi *machine learning* tersebut memanfaatkan teknik yang disebut *deep learning* [6]. *Conventional machine learning* memiliki keterbatasan dalam kemampuannya untuk memproses data-data dalam bentuk mentah. Membangun sebuah sistem *machine learning* membutuhkan keahlian dan rekayasa yang cermat dalam membuat fitur ekstraksi untuk mengubah data yang mentah menjadi suatu representasi yang diperlukan. *Representation learning* memungkinkan mesin dapat secara otomatis mengolah data mentah untuk menemukan representasi yang diperlukan dalam melakukan deteksi atau klasifikasi. Metode *deep learning* juga dapat disebut sebagai metode *representation learning* yang memiliki beberapa tingkat representasi. *Deep learning* memungkinkan model komputasi yang terdiri dari beberapa lapisan proses pengolahan untuk dapat mempelajari representasi data melalui beberapa tingkat abstraksi. Metode ini secara drastis meningkatkan pengembangan dalam bidang pengenalan suara (*voice recognition*), pengenalan objek visual (*image recognition*), deteksi objek (*object detection*) dan begitu banyak hal lainnya seperti dalam penemuan obat dan dalam bidang genomika. Saat ini *Deep learning* telah membawa terobosan dalam berbagai bidang *artificial intelligence*. *Deep learning* dapat menemukan struktur yang begitu rumit di dalam sebuah datasheet yang sangat besar dengan menggunakan algoritma *backpropagation* [7].

2.4 Computer Vision

Computer Vision adalah suatu ilmu yang mempelajari tentang cara pandang komputer terhadap objek disekelilingnya dengan mampu menganalisisnya. *Computer Vision* merupakan gabungan dari *digital image processing*, *pattern recognition*, *computer graphics* dan *machine learning*. *Computer Vision* berada dalam naungan ilmu komputer yang mencakup algoritma, pemrosesan data dan grafik, ilmu fisika yang mencakup optik dan sensor, ilmu matematika yang mencakup kalkulus dan teori informasi, serta ilmu biologi yang mencakup visual stimuli dan *neural processing*. *Computer vision* juga merupakan kombinasi antara sistem pencahayaan (*lighting system*) dan analisa citra atau gambar (*image analysis*). Otak manusia dapat mengenali wajah sesama manusia meskipun baru pertama kali bertemu dan dapat mengenali objek dengan cepat. Bagi komputer, sebuah gambar hanya merupakan sekumpulan pixel. *Computer Vision* bertujuan untuk mengenali sebuah gambar dengan baik, sehingga dapat menghasilkan informasi-informasi yang penting di dalam sebuah gambar. Dengan kata lain tujuan *computer vision* adalah untuk mengajarkan komputer bisa membuat pixel-pixel menjadi hidup dan terasa seperti di dunia nyata.

2.5 Convolutional Neural Network

Ciri pendefinisian penting dari *convolutional neural network* ialah operasi, yang diklaim sebagai *convolution*. Operasi *convolution* artinya operasi titik-produk antara set bobot struktur-grid dan input struktur-grid serupa yang diambil berasal lokasi spasial tidak selaras dalam volume input. Jenis operasi ini

berguna buat data dengan tingkat spasial tinggi atau lokalitas lain, seperti data gambar [8]. *Convolution* artinya proses sederhana pada mana kita menerapkan matriks (pula diklaim kernel atau filter) ke gambar sehingga kita dapat memperkecil ukurannya, atau menambahkan beberapa lapisan pengisi supaya ukurannya tetap sama. *Convolution* juga digunakan buat mengekstraksi fitur khusus dari suatu gambar, mirip bentuk, tepi, dan sebagainya [9].

2.6 You Only Look Once v3

Terinovasi dari ide-ide para komunitas penelitian computer vision, YOLO v3 merupakan versi ketiga dari algoritma object detection YOLO yang mulai diperkenalkan pada tahun 2015 oleh Joseph Redmon [10]. Dengan menerapkan jaringan saraf tunggal, algoritma dari YOLO v3 akan membagi suatu gambar ke dalam bentuk grid dengan ukuran yang telah ditentukan dan untuk setiap cell akan dilakukan prediksi tiga *bounding boxes* dengan *class probabilities*. Namun, YOLO v3 akan melakukan prediksi *bounding box* pada tiga ukuran/skala grid yang berbeda, untuk itu setiap cell pada akhirnya akan memprediksi sembilan *bounding boxes* dengan *class probabilities*. *Bounding boxes* yang memiliki *score* di atas batas nilai *confidence threshold* dan di atas batas nilai *IoU threshold* dikategorikan sebagai sebuah objek dengan sebuah *probability class*.

3. METODE PENELITIAN

3.1. Data Yang Digunakan

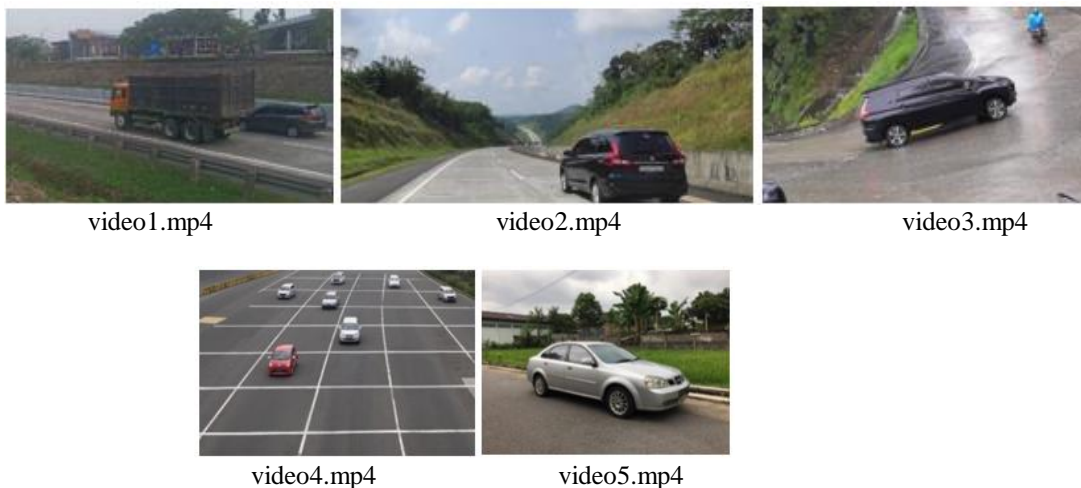
Terdapat 5 file video yang berisi rekaman lalu lintas yang digunakan sebagai bahan untuk menguji sistem yang dibangun dan sebuah pretrained weights dari YOLO v3 yang berisi hasil training terhadap objek-objek yang hendak dideteksi.

3.1.1 Video Uji Coba

Video uji coba yang pertama diambil dari perekaman langsung di jalan dan video uji coba yang kedua diambil dari perekaman langsung di jalan. Video uji coba yang ketiga didapatkan dari sebuah CCTV yang ada di jalan dengan kualitas yang rendah untuk menguji kualitas dari video. Untuk video uji coba yang keempat dan video uji coba yang kelima didapatkan dari sebuah CCTV yang sama yang ada di jalan pada waktu yang berbeda dalam rangka menguji tingkat kecahayaan dari video. Informasi dari masing-masing video uji coba dapat dilihat pada Tabel 1 dan sudut pengambilan masing-masing video uji coba dapat dilihat pada Gambar 1.

Tabel 1. Video Uji Coba

Nama File	Durasi	Total Frame	Jumlah Kendaraan				Jumlah
			Mobil	Truk	Bus	S.Motor	
Video1	00:50	1500	15	10	2	10	37
Video2	01:00	1807	14	10	2	21	47
Video3	00:10	250	4	1	1	14	20
Video4	00:20	481	10	1	0	15	26
Video5	00:15	360	8	2	0	5	15



Gambar 1. Sudut Pengambilan Video Uji Coba

3.1.2. Pretrained Weights

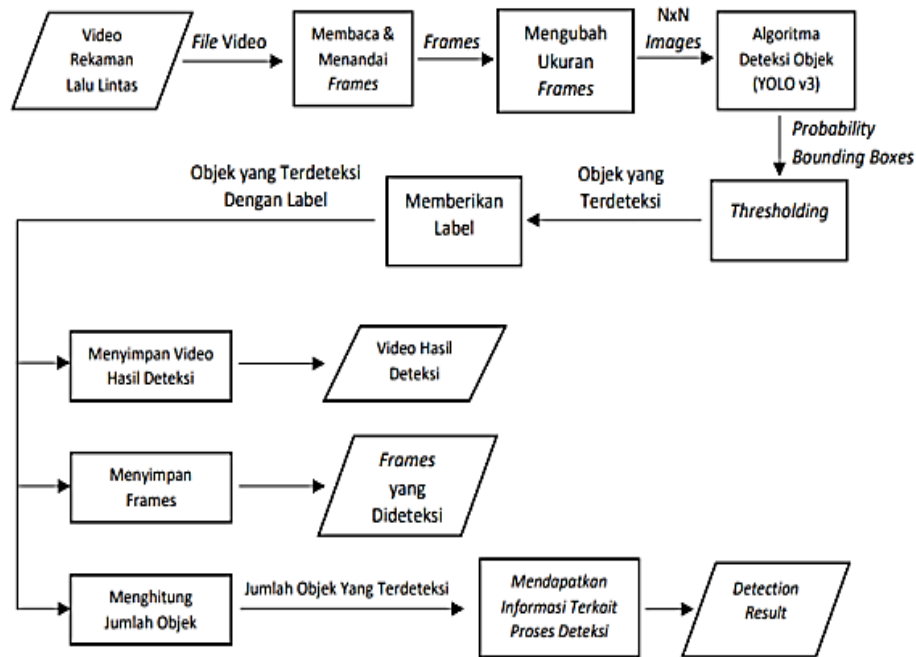
Pretrained weights dari YOLO v3 diunduh dari website official YOLO. *Pretrained weights* dari YOLO v3 dilatih menggunakan dataset yang diambil dari Google Open Images Dataset, dalam arti dataset yang digunakan oleh YOLO bukan bersifat lokal, sehingga dapat dikatakan bahwa ada beberapa kendaraan khususnya pada jenis kendaraan mobil dan truk yang ada pada dataset yang digunakan oleh YOLO sedikit berbeda dengan biasa yang kita kenal dalam versi lokal. Bentuk-bentuk objek kendaraan mobil, truk, bus dan motor dapat dilihat pada Tabel 2.

Tabel 2. Bentuk Objek Kendaraan

Jenis Kendaraan	Contoh Gambar
Mobil	
Truk	
Bus	
Sepeda Motor	

3.2. Rancangan Aplikasi

Proses kerja dari sistem yang dibangun mulai dari pemrosesan input sampai pada output yang dihasilkan dapat dilihat pada *block diagram* yang ada pada Gambar 2. Pada awalnya terdapat sebuah input yaitu berkas video yang berisi rekaman lalu lintas suatu jalan. Selanjutnya terdapat sebuah proses untuk membaca seluruh frame yang membangun input video dan untuk menandai frame yang hendak dilakukan proses deteksi. Untuk melakukan kedua hal tersebut, maka diperlukan sebuah *library Open CV*. *Frame* yang telah ditandai untuk dilakukan proses deteksi pertama-tama dilakukan perubahan ukuran gambar sesuai dengan ukuran yang telah ditentukan dan kemudian dilakukan proses deteksi. *Library OpenCV* kembali digunakan untuk mengubah ukuran dari suatu frame dan untuk melakukan proses deteksi dibutuhkan *framework Mask R-CNN object detection* dan pretrained model dari YOLO v3. Proses deteksi menghasilkan prediksi bounding boxes pada setiap frame yang telah ditandai. Seluruh bounding box yang terprediksi kemudian dilakukan *confidence thresholding* dan *intersection over union thresholding* untuk menghasilkan *bounding box* yang bertanggung jawab terhadap objek sebenarnya yang ada di dalam frame atau dengan kata lain proses thresholding menghasilkan objek yang dinyatakan terdeteksi. Selanjutnya, objek yang dinyatakan terdeteksi dilakukan proses pemberian label. Dalam proses pemberian label digunakan beberapa tambahan library, seperti *library Pillow* dan *library Seaborn* untuk menggambar persegi pada objek yang dinyatakan terdeteksi dan untuk menambahkan teks label pada setiap persegi. Pada akhirnya terdapat 3 proses, yaitu proses menghitung jumlah objek yang dinyatakan terdeteksi pada setiap frame yang telah ditandai, proses untuk menyimpan video hasil deteksi ke dalam bentuk berkas video yang baru dan proses untuk menyimpan frame yang telah ditandai yang telah dilakukan proses deteksi ke dalam bentuk berkas gambar. Proses menghitung jumlah objek yang telah dinyatakan terdeteksi menghasilkan jumlah kendaraan yang terdeteksi pada masing-masing kategori, di mana terdapat empat kategori yaitu, kategori kendaraan mobil, truk, bus dan sepeda motor. Hasil tersebut kemudian dibawa ke proses selanjutnya untuk ditambahkan dengan informasi terkait proses deteksi, seperti id deteksi, tanggal dan waktu deteksi, nama berkas video yang dideteksi, serta durasi dari berkas video yang dideteksi.



Gambar 2. Diagram Blok Prinsip Kerja Sistem

3.3 Teknik Evaluasi

Evaluasi di dalam penelitian ini ialah suatu aktivitas untuk mengusut kinerja asal sistem yang sudah dibangun, di mana sistem yang dibangun di dalam penelitian ini mulai dari input sampai pada output yang dihasilkan hanya menggunakan pemrosesan CPU dengan prosesor AMD A9-9425. Sistem yang dibangun dipasang dalam sistem operasi windows 10 64-bit dengan 4GB kapasitas RAM.

Pada proses evaluasi dilakukan pengukuran secara manual terhadap kecepatan proses deteksi dan perhitungan jumlah kendaraan yang terdeteksi berdasarkan klasifikasi jenis kendaraan mobil, truk, bus dan sepeda motor. Di dalam proses evaluasi juga akan diukur tingkat akurasi deteksi objek dalam mendeteksi dan mengklasifikasi objek kendaraan. Untuk mengukur tingkat akurasi deteksi objek, maka akan dihitung berapa banyak jumlah objek kendaraan yang terdeteksi benar dari keseluruhan objek yang terdeteksi oleh sistem. Pengukuran taraf akurasi deteksi objek digambarkan dalam bentuk tabel confusion matrix 5x5. Berikut merupakan parameter-parameter yang dipergunakan pada pengukuran tingkat akurasi deteksi objek:

3.3.1 True Positive (TP)

True Positive atau biasa disingkat dengan TP merupakan objek sebenarnya yang terdeteksi benar oleh sistem.

3.3.2 False Positive (FP)

False Positive atau biasa disingkat dengan FP merupakan noise yang terdeteksi oleh sistem sebagai objek.

3.3.3 False Negative (FN)

False Negative atau biasa disingkat dengan FN merupakan objek sebenarnya yang tidak terdeteksi oleh sistem.

3.3.4 Accuracy

Accuracy adalah hasil perhitungan tingkat akurasi object detection terhadap objek kendaraan pada keseluruhan frame yang dilakukan proses deteksi. Rumus dari perhitungan accuracy dapat dilihat pada persamaan (1).

$$Accuracy = \frac{Jumlah\ Objek\ Yang\ Terdeteksi\ Benar}{Jumlah\ Keseluruhan\ Objek\ Yang\ Terdeteksi} \quad (1)$$

3.3.5 Precision

Precision adalah jumlah prediksi yang benar dibandingkan keseluruhan hasil yang terprediksi oleh sistem. Dalam hal ini *precision* akan menjawab berapa jumlah objek yang terdeteksi benar dari keseluruhan jumlah objek yang terdeteksi oleh sistem. Rumus dari perhitungan precision dapat dilihat pada persamaan (2).

$$Precision = \frac{TP}{(TP + FP)} \tag{2}$$

3.3.6 Recall

Recall adalah jumlah prediksi yang benar dibandingkan dengan keseluruhan hasil yang sebenarnya. Dalam hal ini recall akan menjawab berapa jumlah objek yang terdeteksi benar dari keseluruhan jumlah objek sebenarnya. Rumus dari perhitungan recall dapat dilihat pada persamaan (3).

$$Recall = \frac{TP}{(TP + FN)} \tag{3}$$

4. HASIL DAN PEMBAHASAN

4.1 Tampilan Tatap Muka Sistem

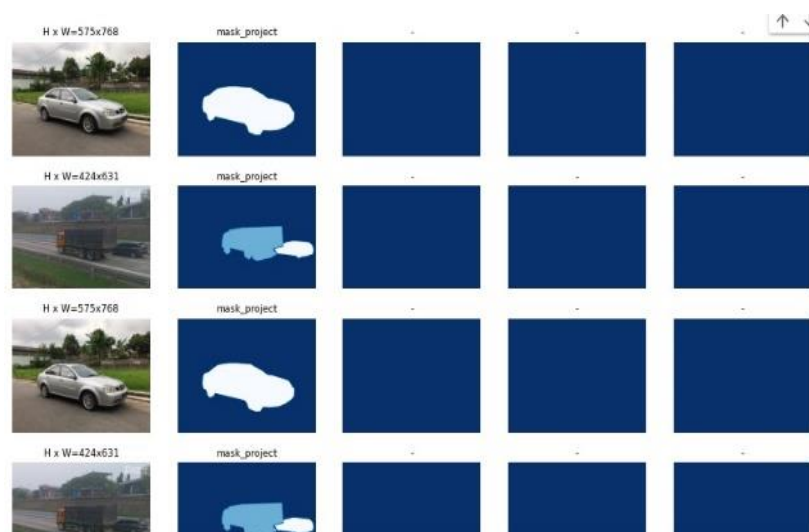
Tampilan tatap muka adalah tampilan visual dari suatu sistem untuk membuat user dapat berinteraksi dengan sistem dengan mudah. Tampilan tatap muka yang dibangun dalam penelitian ini berbasis web. Tampilan halaman utama sistem dapat dilihat pada Gambar 3, sedangkan tampilan halaman untuk memutar video hasil deteksi dan untuk melihat frame yang telah ditandai dapat dilihat pada Gambar 4 dan 5.



Gambar 3. Antar Muka Halaman Utama



Gambar 4. Antar Muka Untuk Memutar Video Hasil Deteksi



Gambar 5. Antar Muka Untuk Menampilkan Frame Yang Telah Ditandai.

4.2 Evaluasi Sistem

Berikut merupakan hasil evaluasi kecepatan proses deteksi, jumlah kendaraan yang terdeteksi, serta tingkat akurasi deteksi objek dalam mendeteksi dan mengklasifikasi objek kendaraan:

4.2.1 Kecepatan Proses Deteksi

Telah dilakukan pengujian sebanyak dua kali secara manual untuk menghitung kecepatan proses deteksi pada kelima file video uji coba dengan durasi dan total frame yang berbeda yang dapat dilihat pada Tabel 3.

Tabel 3. Kecepatan Proses Deteksi

Nama File	Durasi (m:d)	Total Frame	Kecepatan (m:d)(1)	Kecepatan (m:d)(2)
Video1	00:50	1500	04:25	04:05
Video2	01:00	1807	04:38	04:29
Video3	00:10	250	01:36	01:19
Video4	00:20	481	01:59	01:44
Video5	00:15	360	10:43	01:31

Kecepatan proses deteksi menyatakan bahwa semakin banyak frame yang akan dideteksi menunjukkan kecepatan proses deteksi lebih lama, namun kecepatan proses deteksi tidak menentu pada saat dilakukan dua kali pengujian pada file video yang sama.

4.2.2 Jumlah Kendaraan Yang Terdeteksi

Hasil perhitungan jumlah kendaraan yang terdeteksi pada kelima file video yang digunakan sebagai video uji coba berdasarkan klasifikasi mobil, truk, bus dan sepeda motor dapat dilihat dalam Tabel 4. Setiap video uji coba memiliki tiga parameter pada masing-masing class, yaitu parameter yang menyimpan jumlah kendaraan sebenarnya yang ada di dalam video (Act), parameter yang menyimpan jumlah kendaraan yang ada di dalam seluruh frame yang dilakukan proses deteksi (In) dan parameter yang menyimpan jumlah kendaraan yang terdeteksi oleh sistem (Pr).

Konsistensi jumlah kendaraan sebenarnya yang ada di dalam video dan jumlah kendaraan yang ada di dalam frame pada tabel 4 kurang efektif dikarenakan teknik dalam mendeteksi dan menghitung objek yang ada pada kelima file video tidak dilakukan pada seluruh frame di dalam video melainkan hanya dilakukan setiap interval 5 detik pada durasi video. Dengan demikian ada beberapa objek yang terlewatkan atau yang tidak berada di dalam frame dan bahkan ada beberapa objek yang sama yang sudah berada di dalam dua frame. Untuk itu, selanjutnya dilakukan perhitungan secara manual terhadap objek-objek yang tidak berada di dalam frame dan objek yang sama yang sudah berada di dalam dua frame, serta objek-objek yang tertutupi oleh objek yang lain yang membuat sistem deteksi tidak dapat mendeteksinya, hasil perhitungan tersebut dapat dilihat pada Tabel 5.

Tabel 4. Hasil Perhitungan Jumlah Kendaraan

Nama File	Mobil			Truk			Bus			Sepeda Motor			Overall		
	Act	In	Pr	Act	In	Pr	Act	In	Pr	Act	In	Pr	Act	In	Pr
Video1	15	5	5	10	8	7	2	2	2	10	1	1	37	16	15
Video2	14	10	11	10	8	7	2	2	2	21	11	9	47	31	29
Video3	4	3	2	1	1	2	1	1	1	14	9	0	20	14	5
Video4	10	8	4	1	1	0	0	0	0	15	11	0	26	20	4
Video5	8	7	7	2	2	4	0	0	0	5	3	3	15	12	14

Tabel 5. Jumlah Kendaraan Yang Tidak Ada Dalam Frame

Nama File	Keseluruhan Jumlah Objek	Objek Yang Ada Dalam Frame	Objek Yang Terulang	Objek Yang Tidak Ada dalam Frame	Objek Yang Tersembunyi
Video1	37	16	0	21	0
Video2	47	31	1	15	2
Video3	20	14	0	4	2
Video4	26	20	0	6	0
Video5	15	12	0	3	0

Berdasarkan hasil perhitungan jumlah kendaraan yang lewat dapat dikatakan bahwa pengambilan sudut dan kualitas rekaman video lalu lintas berpengaruh dalam perhitungan objek kendaraan. Hal ini dikarenakan jika pengambilan sudut terlalu rendah, maka akan terdapat objek kendaraan yang tertutupi oleh objek kendaraan yang lain dan jika kualitas rekaman video lalu lintas terlalu rendah maka akan ada objek yang tidak dapat terlihat jelas oleh sistem deteksi. Tingkat kecerahan cahaya juga berpengaruh dalam sistem deteksi di mana telah dilakukan pengujian terhadap file video4.mp4 dan file video5.mp4, di mana file video4.mp4 dan file video5.mp4 memiliki lokasi dan pengambilan sudut yang sama, namun pengujian tersebut memiliki hasil perhitungan objek yang rendah pada waktu malam hari, dalam hal ini khususnya objek kendaraan sepeda motor sulit untuk dideteksi oleh sistem.

4.3.3 Tingkat Akurasi Deteksi Objek

Hasil pengujian tingkat akurasi deteksi objek yang dilakukan pada kelima file video yang digunakan sebagai video uji coba dapat dilihat pada Tabel 6. Rata-rata tingkat akurasi deteksi objek pada pengujian kelima file video uji coba adalah 90.8%. Nilai overall accuracy yang tinggi belum menjamin ketepatan dari algoritma deteksi objek yang digunakan dalam melakukan deteksi dan klasifikasi, hal ini dapat dilihat dari nilai precision dan nilai recall setiap class. Misalkan, meskipun nilai overall accuracy yang ada pada file video4.mp4 adalah 100%, namun file video4.mp4 memiliki nilai recall yang rendah pada masing-masing class dikarenakan ada beberapa objek yang tidak dapat terdeteksi oleh algoritma deteksi objek. Berdasarkan perhitungan sebelumnya mengenai jumlah kendaraan pada masing-masing class menyatakan bahwa algoritma deteksi objek sulit untuk mendeteksi objek yang ada pada video yang memiliki kualitas rendah atau yang memiliki tingkat kecahayaan yang kurang.

Tabel 6. Hasil Pengujian Tingkat Akurasi Deteksi Objek

Nama File	Mobil		Truk		Bus		Sepeda Motor		Overall Accuracy
	P	R	P	R	P	R	P	R	
Video1	100%	100%	100%	87%	100%	100%	100%	100%	100%
Video2	91%	100%	100%	87%	100%	100%	100%	81%	96%
Video3	100%	66%	50%	100%	100%	100%	N/A	0%	80%
Video4	100%	50%	N/A	0%	N/A	N/A	N/A	0%	100%
Video5	100%	87%	25%	100%	N/A	N/A	100%	100%	78%

5. KESIMPULAN

Berdasarkan hasil penelitian dan pengujian terhadap sistem yang telah dibangun, maka diambil kesimpulan bahwa Framework Mask R-CNN Object Detection dapat diimplementasikan untuk membangun sebuah sistem yang dapat mendeteksi dan menghitung jumlah kendaraan berdasarkan jenis kendaraan yang melewati suatu jalan. Pengujian tingkat akurasi dalam mendeteksi objek kendaraan berdasarkan klasifikasi jenis kendaraan yang dilakukan pada kelima file video uji coba memiliki nilai rata-rata, yaitu 90.8% akurat dan perhitungan jumlah kendaraan berdasarkan klasifikasi jenis kendaraan yang melewati suatu jalan yang dilakukan setiap interval 5 detik di dalam video kurang efektif untuk diterapkan.

REFERENSI

- [1] W. Supriyatin, Y. Rafsyam, dan Jonifan, "Analisis Pelacakan Objek Menggunakan Background Estimation Pada Kamera Diam Dan Bergerak," *Orbith Maj. Ilm. Pengemb. Rekayasa dan Sos.*, vol. 13, no. 2, hal. 124–130, 2017.
- [2] W. Supriyatin, "Analisis Perbandingan Pelacakan Objek Menggunakan Optical Flow Dan Background Estimation Pada Kamera Bergerak," *Ilk. J. Ilm.*, vol. 11, no. 3, hal. 191–199, 2019, doi: 10.33096/ilkom.v11i3.452.191-199.
- [3] I. A. Dewi, L. Kristiana, A. R. Darlis, Dan R. F. Dwiputra, "Deep Learning RetinaNet based Car Detection for Smart Transportation Network," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 7, no. 3, hal. 570–584, 2019, doi: 10.26760/elkomika.v7i3.570.
- [4] A. A. Firmansyah, R. Effendie, dan A. Santoso, "Deteksi Halangan Menggunakan Metode Stereo R-CNN pada Mobil Otonom," *J. Tek. ITS*, vol. 9, no. 2, hal. E160–E166, 2020.
- [5] Imanuddin, F. Alhadi, R. Oktafian, dan A. Ihsan, "Deteksi Mata Mengantuk pada Pengemudi Mobil Menggunakan Metode Viola Jones," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, hal. 321–329, 2019, doi: 10.30812/matrik.v18i2.389.
- [6] D. J. P. Manajang, S. R. U. A. Sompie, dan A. Jacobus, "Implementasi Framework Tensorflow Object Detection API Dalam Mengklasifikasi Jenis Kendaraan Bermotor," *J. Tek. Inform.*, vol. 15, no. 3, hal.

- 171–178, 2020, doi: 10.35793/jti.15.3.2020.29775.
- [7] Y. Lecun, Y. Bengio, dan G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, hal. 436–444, 2015, doi: 10.1038/nature14539.
- [8] C. C. Aggarwal, *Neural Networks and Deep Learning*. Switzerland: Springer, 2018.
- [9] H. Singh, *Practical Machine Learning and Image Processing*. New York: Apress, 2019.
- [10] J. Redmon dan A. Farhadi, “YOLOv3: An Incremental Improvement,” 2018.