

# PERANCANGAN PROGRAM PENDETEKSI DAN PENGKLASIFIKASI JENIS KENDARAAN DENGAN METODE *CONVOLUTIONAL NEURAL NETWORK (CNN) DEEP LEARNING*

Ridwan Gunadi Fajri \*), Imam Santoso dan Yosua Alvin Adi Soetrisno.

Departemen Teknik Elektro, Universitas Diponegoro  
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

\*)ridwangf24@gmail.com

## Abstrak

Lalu lintas memerlukan teknologi yang handal dalam mendeteksi dan menghitung jumlah kendaraan. Hal ini berguna untuk mengurangi kemacetan dan mentertibkan lalu lintas. Pada penelitian ini, dirancang suatu program pendeteksi, pengklasifikasi, dan penghitung kendaraan menggunakan metode *Convolutional Neural Network (CNN) Deep Learning* dengan algoritma YOLO. YOLO adalah sebuah algoritma CNN yang dikembangkan untuk mendeteksi suatu objek secara *real-time*. Masukan sistem adalah citra kendaraan yang akan dideteksi. Sistem akan mendeteksi kendaraan, dipisahkan berdasarkan jenisnya, dan dihitung total kendaraannya. Nilai hasil perhitungan program akan dikirimkan ke mikrokontroler sebagai pengatur durasi lampu lalu lintas dan juga disimpan di dalam *database* agar selalu tercatat setiap perhitungannya. Keakuratan program didapat dengan membandingkan hasil pendeteksian, pengklasifikasian, dan perhitungan kendaraan dengan nilai perhitungan manual. Pada pengujian berdasarkan kelas, didapat nilai keakuratan program sebesar 91,4%. Pada pengujian berdasarkan pengaruh faktor lingkungan, pada kondisi hujan didapatkan nilai akurasi sebesar 88,4%, pada kondisi berkabut sebesar 70%, dan kondisi malam hari sebesar 78,2%. Pada pengujian komunikasi data, program dapat mengirim sinyal ke mikrokontroler dan file *database* secara sempurna.

**Kata kunci :** Deteksi kendaraan, CNN, Deep Learning, YOLO

## Abstract

Traffic requires reliable technology in detecting and counting the number of vehicles. This is useful to reduce traffic congestion and order traffic. In this final project, a vehicle detection, classifier, and vehicle counter program is designed using the Deep Learning Convolutional Neural Network (CNN) method with YOLO algorithm. YOLO is a CNN algorithm developed to detect an object in real time. The input system is the image of the vehicle to be detected. The system will detects vehicle, separated by type, and calculated the total vehicle. The value of the calculation program will be sent to the microcontroller as a regulator of the duration of the traffic lights and also stored in the database so that it is always recorded every calculation. The accuracy of the program is obtained by comparing the results of detection, classification and calculation of vehicles with the value of manual calculations. In testing based on class, the accuracy of the program is 91,4%. In testing based on the influence of natural factors, the rain conditions obtained an accuracy value of 88,4%, foggy conditions by 70%, and night conditions by 78,2%. In testing data communication, the program can send signals to the microcontroller and database file perfectly.

**Keywords:** Vehicle Detection, Convolutinal Neural Network (CNN), Deep Learning, YOLO

## 1. Pendahuluan

Teknologi pengolahan citra digital dan *computer vision* telah berkembang sangat pesat dan banyak diterapkan untuk berbagai kepentingan, diantaranya adalah penerapan pada sektor lalu lintas. Pemasangan kamera CCTV (*Closed-Circuit Television*) di beberapa ruas jalan perkotaan memungkinkan petugas untuk mencegah tindak pelanggaran, memantau arus lalu lintas dan mengamati tingkat kepadatan pengguna kendaraan. Namun, penggunaan sistem kamera CCTV hanya berfungsi

sebagai sistem pemantauan pasif saja [1]. Oleh karena itu, dibutuhkan teknologi yang dapat mendeteksi kendaraan secara langsung dan melakukan aksi secara otomatis terhadap kendaraan yang terdeteksi seperti menghitung kepadatan lalu lintas atau mendeteksi pelanggaran lalu lintas.

Berbagai macam metode pada pengolahan citra digital telah banyak dikembangkan salah satunya adalah *Convolutional Neural Network (CNN)*. Konsep CNN menggunakan teknologi kecerdasan buatan yang telah

dilatih menggunakan data dalam jumlah besar dan algoritma tertentu yang membuatnya dapat belajar dari data yang telah diperoleh [2]. CNN adalah salah satu contoh dari penerapan *deep learning* yang banyak diminati dan diterapkan didalam penelitian karena poses pembelajarannya memiliki karakter yang mirip dengan pembelajaran pada manusia [3]

Penelitian ini bertujuan untuk merancang sistem yang dapat melakukan pengenalan citra kendaraan menggunakan metode *convolutional neural network* tipe *deep learning* yang diklasifikasikan dalam empat kelas, yaitu bis, mobil, sepeda motor, dan truk. Jenis citra yang diamati hanya citra diam. Algoritma yang digunakan pada pengenalan kendaraan adalah YOLO (*You Only Look Once*), yaitu sebuah algoritma CNN yang dikembangkan untuk mendeteksi suatu objek secara *real-time*

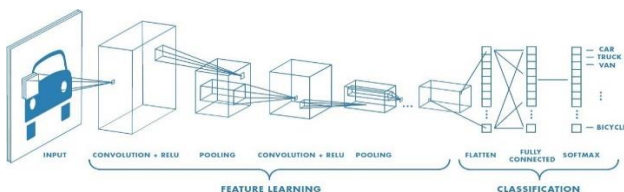
Citra adalah jenis multimedia yang memiliki peran besar untuk penyajian informasi dalam bentuk visual. Sebuah riset yang mengacu pada pengolahan citra menggunakan perhitungan secara digital disebut dengan pengolahan citra digital [4]. Tujuan dari proses citra digital adalah agar suatu citra dapat diolah menggunakan komputer atau perangkat digital lainnya dan memperbaiki kualitas citra sehingga mudah diartikan oleh manusia [5].

Pengolahan citra digital adalah proses mengolah suatu citra agar sesuai dengan keinginan dan kepentingan kita. Pengolahan citra digital mencakup proses mendeteksi dan mengklasifikasi objek pada suatu citra, perbaikan kualitas citra, pemberian tanda air, dan sebagainya.

## 2. Dasar Teori

### 2.1. Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) adalah salah satu jenis *neural network* yang biasa digunakan pada data citra [6]. CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah gambar. CNN terdiri dari sejumlah interkoneksi node komputasi yang disebut dengan neuron. CNN umumnya terdiri dari tiga *layer* utama, yaitu *input layer*, *hidden layer* dan *output layer*. CNN belajar langsung melalui data citra, sehingga dapat menghilangkan ekstraksi ciri secara manual [7]. Berikut merupakan gambaran umum arsitektur *Convolution Net*.



Gambar 1. Gambaran umum proses konvolusi

Secara umum tipe lapisan CNN dibagi menjadi dua bagian, yaitu :

#### a) Layer Ekstraksi Fitur (*feature extraction layer*)

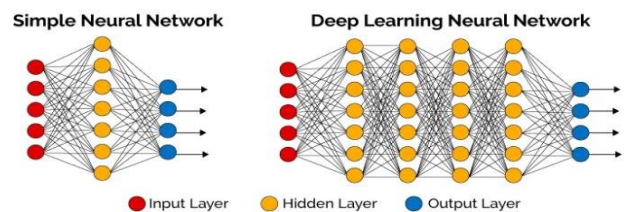
Layer ini letaknya ada di awal arsitektur yang tersusun atas beberapa lapisan dan di setiap susunan lapisannya atas neuron yang terkoneksi pada *local region* dari lapisan sebelumnya. Lapisan pada jenis pertama yaitu adalah *convolutional layer* yang berfungsi sebagai layer untuk proses operasi konvolusi dan lapisan kedua adalah *pooling* yang berfungsi untuk mereduksi input secara spasial (mengurangi jumlah parameter) dengan operasi *down-sampling* [8].

#### b) Layer Klasifikasi (*classification layer*)

Layer ini tersusun atas beberapa lapisan yang di setiap lapisan tersusun atas neuron yang terkoneksi secara penuh (*fully connected*) dengan lapisan yang lainnya. Layer ini menerima input dari hasil output layer ekstraksi fitur gambar berupa vektor yang kemudian di transformasikan *Network* dengan tambahan beberapa *hidden layer* [9].

## 2.2. Deep Learning

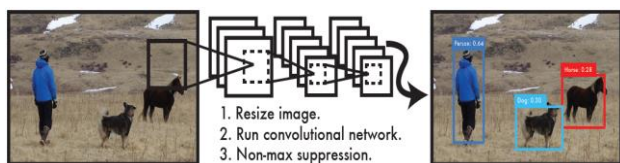
*Deep Learning* merupakan bidang studi *machine learning* yang memanfaatkan banyak *layer* sebagai pengolah informasi untuk melakukan ekstraksi fitur, pengenalan pola dan klasifikasi [10]. Metode *deep learning* memanfaatkan arsitektur CNN sebagai implementasi untuk permasalahan yang berkaitan dengan dataset dalam jumlah besar [11].



Gambar 2. Perbandingan *Layer* pada Jaringan Syaraf Sederhana dengan *Deep Learning*

## 2.3. YOLO

YOLO (*You Only Look Once*) adalah sebuah algoritma yang dikembangkan untuk mendeteksi suatu objek secara *real-time*. YOLO membagi citra masukan menjadi suatu *grid* berukuran  $S \times S$  dimana tiap sel dari *grid* tersebut akan memprediksi *bounding box* dan menghasilkan nilai untuk tiap kelas. Masing-masing *bounding box* terdiri dari lima nilai prediksi yaitu pusat koordinat  $x$ , pusat koordinat  $y$ , lebar sel ( $w$ ), tinggi sel ( $h$ ) dan nilai *confidence* [12].



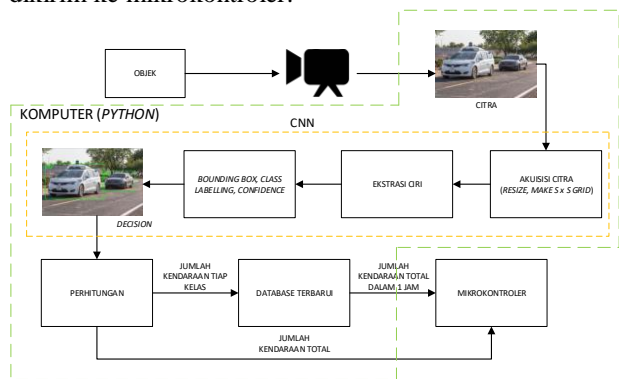
Gambar 3. Sistem deteksi Yolo [12]

Pada pembagian citra masukan menjadi ukuran *grid*  $S \times S$ , bagian pusat yang menjadi titik tengah dari *grid cell* bertanggung jawab untuk proses pendeteksian objek [13]. Masing-masing *grid cell* memprediksi kotak pembatas dan nilai *confidence* untuk setiap kotak. Nilai ini menyatakan berapa besar tingkat keyakinan sistem terhadap kotak pembatas yang mengandung inti informasi objek dan juga seberapa akurat *grid cell* dalam memprediksi kotak pembatas tersebut. Secara matematis nilai keyakinan dapat dinyatakan sebagai berikut:

$$Confidence\ Score = P_r(Object) * IOU_{pred}^{truth} \quad (1)$$

### 3. Metode

Pada gambar 4, dapat dilihat diagram blok dalam perangan sistem pendeteksi dan pengklasifikasi. Berdasarkan gambar 4, dapat dilihat bahwa pada tahap awal, citra akan diakuisisi oleh algoritma YOLO. Kemudian citra tersebut diekstraksi cirinya dan didapat nilai  $x, y, w, h$ , dan nilai *confidence*-nya. Apabila sudah dihitung, maka akan tampil *bounding box*, label kelas, dan nilai *confidence* nya. Proses tersebut dinamakan *decision*. Proses akuisisi hingga *decision* merupakan proses dari *convolutional neural network* (CNN). Seluruh objek yang terdeteksi akan dihitung dan keluarannya adalah akan dikirim ke mikrokontroler.



Gambar 4. Diagram blok sistem pendeteksi dan pengklasifikasi jenis kendaraan

### 3.1. Pengumpulan Data Latih

Data yang digunakan pada penelitian ini berupa kumpulan gambar kendaraan dari empat kelas yakni bis, mobil, sepeda motor, truck, dan juga kumpulan dari keempatnya.

Jumlah data latih yang digunakan pada penelitian ini adalah sebanyak 3282 citra data latih yang terdiri dari 675 gambar bis, 1070 gambar mobil, 700 gambar sepeda motor, 700 gambar truk, dan 137 gambar campuran dari kesemuanya. Data set dapat diperoleh pada *link* berikut [https://drive.google.com/drive/folders/1aNesTRMZ\\_bOAujN7WTftp8LGWSozz1o?usp=sharing](https://drive.google.com/drive/folders/1aNesTRMZ_bOAujN7WTftp8LGWSozz1o?usp=sharing)

### 3.2. Anotasi Citra

Anotasi merupakan proses membuat label dengan cara memberikan kotak batas (*bounding box*) beserta nama kelas pada objek disetiap citra [14]. Proses anotasi citra dilakukan agar data latih memiliki nilai  $x, y, w$ , dan  $h$ . Nilai  $x, y, w$ , dan  $h$  adalah nilai prediksi pada *bounding box* yang berguna untuk proses pelatihan CNN menggunakan algoritma YOLO. Pada proses anotasi, citra yang akan dilatih diberikan label berdasarkan jenis kendaraan yang terlihat. Objek dengan indeks 0 adalah *bus*, indeks 1 adalah *car* (mobil), indeks 2 adalah *motorbike* (sepeda motor), indeks 3 adalah *truck*.

### 3.3. Perancangan CNN

Perancangan CNN merupakan tahapan dimana menyusun sebuah model yang akan digunakan dalam proses pelatihan data [15]. Penyusunan sebuah model CNN terdiri dari menentukan jumlah layer yang akan digunakan, menentukan filter, menentukan fungsi aktivasi dan menentukan ukuran pooling. Jumlah layer yang digunakan adalah 7 layer yang terdiri dari layer *convolutional layer* dan *fully connected layer*, jumlah kelas sebanyak, dan *maximum pooling* dengan filter berukuran  $2 \times 2$ .

### 3.4. Pelatihan Data

Pelatihan data latih dilakukan untuk proses ekstraksi ciri dari masing-masing kelas yang akan diklasifikasikan. Pelatihan data melibatkan seluruh gambar yang ada pada direktori. Jumlah iterasi yang dilakukan pada proses *training* adalah sebanyak 10.000 iterasi. Setelah dilakukan *training* maka akan dihasilkan file bobot dengan format *weights*. Format *weights* adalah format file bobot apabila kita melakukan proses *training* menggunakan *darknet*.

### 3.5. Rancangan Program

Pada perancangan program, bobot dari proses akan diuji menggunakan metode CNN dan didapatkan hasil berupa deteksi, klasifikasi, dan perhitungan kendaraan. Rancangan program dibagi menjadi dua bagian yaitu program pendeteksi dan komunikasi data yang terdiri dari komunikasi *serial* dan database.

a) Program Pendeteksi

Program pendeteksi adalah program yang dirancang untuk menguji keandalan bobot dalam mendeteksi kendaraan menggunakan data uji. Citra data uji akan diproses menggunakan metode CNN dengan parameter referensi data latih berupa bobot hasil *training*. Keluaran dari program ini adalah hasil deteksi dan klasifikasi kendaraan berdasarkan kelas dan perhitungan jumlah kendaraan.

Proses selanjutnya adalah menguji tingkat akurasi bobot menggunakan data uji. Pengujian dilakukan dengan membandingkan hasil perhitungan secara manual untuk menentukan nilai akurasi (*F1*) dengan hasil perhitungan program dengan parameter berupa *true positive (TP)*, *false positive (FP)*, dan *false negative (FN)*. Nilai *FP* dan *FN* merupakan nilai *error* sistem. *TP (True Positive)* merupakan target piksel yang dikenali sebagai objek, *FP (False Positive)* adalah *noise* yang terdeteksi sebagai objek. *Noise* pada penelitian ini yaitu objek selain bus, mobil, sepeda motor, dan truk. *FN (False Negative)* adalah target piksel yang tidak terdeteksi sebagai objek atau target deteksi dengan nilai *confidence* fluktuatif diantara batas nilai ambang (*threshold*) (Lipton dkk, 2014). Untuk menentukan nilai *F1* ditunjukkan pada persamaan berikut.

$$F1\ Score = \left( \frac{TP}{TP+FP+FN} \right) 100\% \quad (2)$$

b) Program Komunikasi Data

Program komunikasi data adalah program yang dirancang untuk menguji apakah program pendeteksi dapat berkomunikasi dengan perangkat lain, kali ini mikrokontroler (*serial*) dan *database*. Terdapat dua jenis program komunikasi data pada program yaitu komunikasi *serial* dan komunikasi *database*

- Komunikasi *Serial*

Komunikasi *serial* adalah komunikasi yang digunakan oleh bahasa pemrograman agar dapat berkomunikasi dengan mikrokontroler. Keluaran dari proses pendeteksian akan dikirimkan ke mikrokontroler. Jumlah kendaraan total akan masuk ke mikrokontroler melalui komunikasi *serial*.

- Komunikasi *Database*

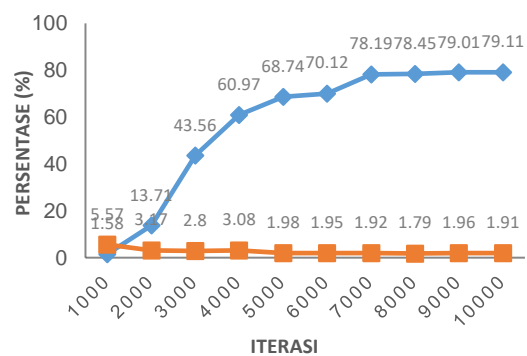
Komunikasi *database* adalah komunikasi yang digunakan oleh bahasa pemrograman agar dapat berkomunikasi dengan file *database*, dengan jenis format *csv*. *Database* diperlukan untuk mencatat dan membaca jumlah kendaraan yang terdeteksi dan nilai tersebut akan selalu terbarukan.

## 4. Pengujian dan Analisis

Pengujian pada program system identifikasi ukuran tubuh dibagi menjadi tiga bagian yaitu analisis hasil *training*, pengujian dan analisis data uji, dan komunikasi data.

### 4.1. Analisis Hasil *Training*

Proses *training* akan menghasilkan file bobot. Hasil bobot dari proses pelatihan ini akan menghasilkan parameter berupa *average loss (avg)* dan *mean average precision (mAP)*.



Gambar 5. Grafik perbandingan nilai mAP dan *avg loss* pada tiap iterasi

Berdasarkan grafik pada gambar 6, nilai mAP selalu naik dan nilai tertinggi didapat pada iterasi ke 10.000 yaitu senilai 79,11%. Pada bobot final atau bobot yang akan digunakan yaitu pada iterasi 10.000, nilai error atau *loss* yang didapatkan adalah senilai 1,91%.

### 4.2. Pengujian dan Analisis Data Uji

Pada bagian ini akan dilakukan proses menguji keakuratan bobot dalam mendeteksi, mengklasifikasi, dan menghitung jumlah kendaraan menggunakan citra masukan pada data uji. Pengujian ini akan dibedakan menjadi dua bagian yaitu citra berdasarkan kelas dan berdasarkan faktor lingkungan. Kelas yang akan diklasifikasi yaitu bus (B), car (C), motorbike (M), dan truck (T) serta total kendaraan (Sum).

a) Pengujian Berdasarkan Kelas

Pengujian ini dilakukan untuk menguji seberapa akurat bobot dari hasil *training* dalam mengenali kendaraan di tiap kelasnya. Gambar hasil pengujian pada kelas pertama (bus) dapat dilihat pada gambar 6 di bawah



Gambar 6. Hasil pengujian kelas pertama

Hasil deteksi, klasifikasi, dan perhitungan pada program dan perhitungan manual pada pengujian kelas pertama dapat dilihat pada tabel 1

Tabel 1. Perbandingan hasil program dan hitung manual untuk data uji kelas pertama

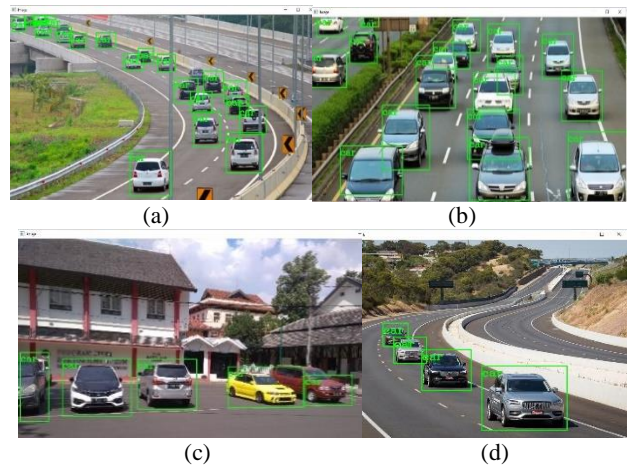
Gambar	Hasil Program					Nilai Hitung Manual				
	B	C	M	T	Sum	B	C	M	T	Sum
a	3	0	0	0	3	3	0	0	0	3
b	4	0	0	0	4	4	0	0	0	4
c	3	0	0	0	3	3	0	0	0	3
d	8	0	0	0	8	6	0	0	0	6

Untuk mendapatkan nilai akurasi, dibutuhkan parameter berupa TP, FP, dan FN. Parameter TP, FP, dan FN dari gambar a, b, c, dan d dapat dilihat pada tabel 2 berikut.

Tabel 2. TP, FP, FN, dan F1 pada data uji kelas pertama

Gambar	TP	FP	FN	F1
a	3	0	0	100%
b	4	0	0	100%
c	3	0	0	100%
d	8	2	0	80%
Total	18	2	0	90% (rata-rata)

Pada tabel 1 gambar d hasil dari program dengan nilai hitung manual menunjukkan hasil yang berbeda. Pada perhitungan oleh program, bus yang terdeteksi sebanyak 8 kendaraan, sementara nilai perhitungan manual sebanyak 6 kendaraan. Perbedaan ini disebabkan karena kurangnya jumlah dan variasi data latih untuk bis dan juga jarak objek dengan objek lain terlalu dekat sehingga ciri kendaraan bis kurang begitu terlihat. Pada tabel 2, hasil program menunjukkan nilai akurasi (F1) rata-rata untuk pengujian kelas pertama yang didapat dari persamaan 2 sebesar 90%. Dapat disimpulkan bobot dari hasil *training* untuk proses uji kelas pertama berjalan dengan baik dan sudah memiliki keandalan yang baik. Untuk gambar hasil pengujian pada kelas kedua (*car*) dapat dilihat pada gambar 7 di bawah



Gambar 7. Hasil pengujian kelas kedua

Hasil deteksi, klasifikasi, dan perhitungan pada program dan perhitungan manual pada pengujian kelas pertama dapat dilihat pada tabel 3.

Tabel 3. Perbandingan hasil program dan hitung manual untuk data uji kelas kedua

Gambar	Hasil Program					Nilai Hitung Manual				
	B	C	M	T	Sum	B	C	M	T	Sum
a	0	22	0	0	22	0	23	0	0	23
b	0	17	0	0	17	0	17	0	0	17
c	0	5	0	0	5	0	5	0	0	5
d	0	4	0	0	4	0	4	0	0	4

Parameter TP, FP, dan FN dari gambar a, b, c, dan d dapat dilihat pada tabel 4 berikut.

Tabel 4. TP, FP, FN, dan F1 pada data uji kelas kedua

Gambar	TP	FP	FN	F1
a	22	0	1	95,6%
b	17	0	0	100%
c	5	0	0	100%
d	4	0	0	100%
Total	48	0	1	97,9% (rata-rata)

Pada tabel 3 gambar a hasil dari program dengan nilai hitung manual menunjukkan hasil yang berbeda yang disebabkan karena posisi mobil yang cukup jauh sehingga ciri mobil kurang terlihat jelas. Pada tabel 4, hasil program menunjukkan nilai akurasi (F1) rata-rata yang didapat dari persamaan 2 sebesar 97,9%. Dapat disimpulkan bobot dari hasil memiliki keandalan yang baik. Untuk gambar hasil pengujian pada kelas ketiga (*motorbike*) dapat dilihat pada gambar 8.





Gambar 8. Hasil pengujian kelas ketiga

Hasil deteksi, klasifikasi, dan perhitungan pada program dan perhitungan manual pada pengujian kelas ketiga dapat dilihat pada tabel 5.

Tabel 5. Perbandingan hasil program dan hitung manual untuk data uji kelas ketiga

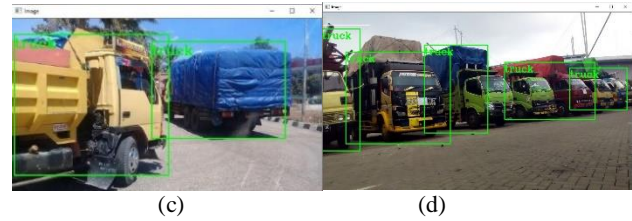
Gambar	Hasil Program					Nilai Hitung Manual				
	B	C	M	T	Sum	B	C	M	T	Sum
a	0	0	2	0	2	0	0	2	0	2
b	0	0	2	0	2	0	0	2	0	2
c	0	0	4	0	4	0	0	4	0	4
d	0	0	8	0	8	0	0	10	0	10

Parameter TP, FP, dan FN dari gambar a, b, c, dan d dapat dilihat pada tabel 6 berikut

Tabel 6. TP, FP, FN, dan F1 pada data uji kelas ketiga

Gambar	TP	FP	FN	F1
a	2	0	0	100%
b	2	0	0	100%
c	4	0	0	100%
d	8	0	2	80%
Total	16	0	2	88,9% (rata-rata)

Pada tabel 5 gambar d hasil dari program dengan nilai hitung manual menunjukkan hasil yang berbeda. Perbedaan ini disebabkan karena kurangnya jumlah dan variasi data latih untuk sepeda motor dan juga jarak objek dengan objek lain terlalu dekat sehingga ciri kendaraan sepeda motor kurang begitu terlihat. Berdasarkan tabel 6, didapatkan nilai akurasi (F1) rata-rata untuk pengujian kelas ketiga yang didapat dari persamaan 2 sebesar 88,9%. Dapat disimpulkan bobot dari hasil *training* untuk proses uji kelas ketiga berjalan dengan baik dan sudah memiliki keandalan yang baik. Untuk gambar hasil pengujian pada kelas keempat (*truck*) dapat dilihat pada gambar 9 di bawah.



Gambar 9. Hasil pengujian kelas keempat

Hasil deteksi, klasifikasi, dan perhitungan pada program dan perhitungan manual pada pengujian kelas ketiga dapat dilihat pada tabel 7

Tabel 7. Perbandingan hasil program dan hitung manual untuk data uji kelas keempat

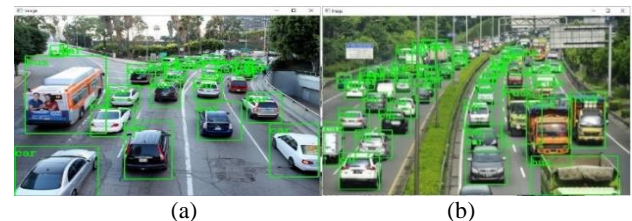
Gambar	Hasil Program					Nilai Hitung Manual				
	B	C	M	T	Sum	B	C	M	T	Sum
a	0	0	0	2	2	0	0	0	2	2
b	0	0	0	3	3	0	0	0	3	3
c	0	0	0	2	2	0	0	0	2	2
d	0	0	0	5	5	0	0	0	7	7

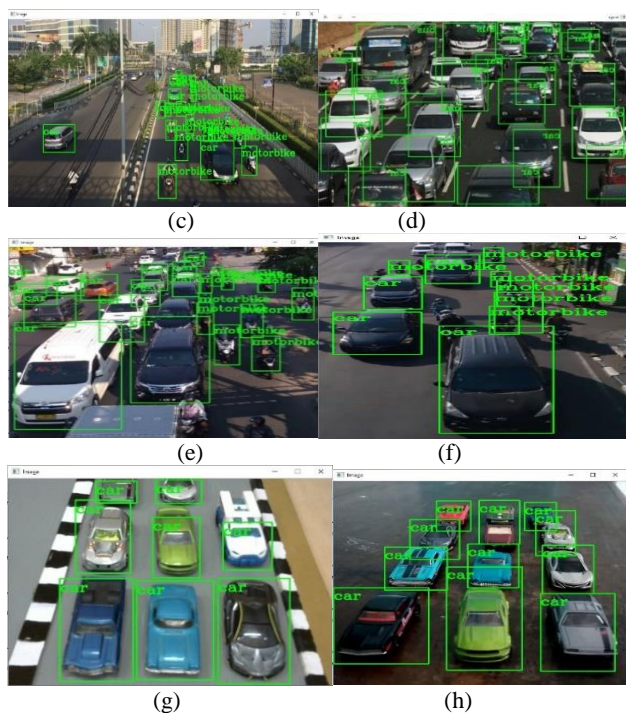
Tabel 7 menampilkan perbandingan hasil program dengan nilai perhitungan manual. Nilai parameter TP, FP, dan FN dari gambar a, b, c, dan d berdasarkan tabel 7 di atas dapat dilihat pada tabel 8 berikut

Tabel 8. TP, FP, FN, dan F1 pada data uji kelas keempat

Gambar	TP	FP	FN	F1
a	2	0	0	100%
b	3	0	0	100%
c	2	0	0	100%
d	5	0	2	71,4
Total	12	0	2	85,7% (rata-rata)

Pada tabel 7 gambar d menunjukkan hasil yang berbeda. Perbedaan ini disebabkan karena kurangnya jumlah dan variasi data latih untuk truk dan juga jarak objek dengan objek lain terlalu dekat sehingga ciri kendaraan truk kurang begitu terlihat. Berdasarkan tabel 8, hasil program menunjukkan nilai akurasi (F1) rata-rata untuk pengujian kelas keempat yang didapat dari persamaan 2 sebesar 85,7%. Dapat disimpulkan bobot dari hasil *training* untuk proses uji kelas ketiga berjalan dengan baik dan sudah memiliki keandalan yang baik. Untuk hasil pengujian pada citra yang terdiri dari gabungan beberapa kelas kendaraan dapat dilihat pada gambar 10 di bawah.





Gambar 10. Hasil pengujian citra gabungan kelas kendaraan

Hasil deteksi, klasifikasi, dan perhitungan pada program dan perhitungan manual pada pengujian data uji untuk citra gabungan kelas kendaraan dapat dilihat pada tabel 9.

Tabel 9. Perbandingan hasil program dan hitung manual untuk data uji citra gabungan kelas kendaraan

Gambar	Hasil Program					Nilai Hitung Manual				
	B	C	M	T	Sum	B	C	M	T	Sum
a	1	26	0	0	27	1	28	0	0	30
b	1	33	0	9	43	1	36	0	10	47
c	0	6	0	13	19	0	6	15	0	21
d	2	23	0	0	25	2	25	0	0	27
e	0	13	10	0	23	0	14	11	0	25
f	0	3	6	0	9	0	5	7	0	12
g	0	8	0	0	8	0	8	0	0	8
h	0	12	0	0	12	0	12	0	0	12

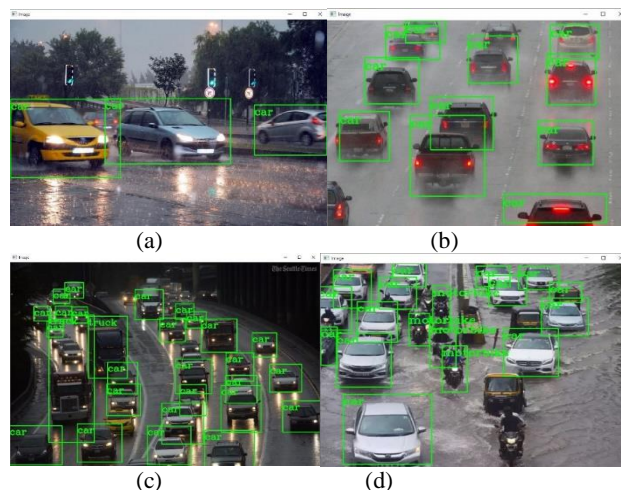
Parameter TP, FP, dan FN dari keseluruhan gambar diatas dapat dilihat pada tabel 10 berikut

Tabel 10. TP, FP, FN, dan F1 pada data uji citra gabungan kelas kendaraan

Gambar	TP	FP	FN	F1
a	27	0	3	93,1%
b	43	0	4	91,4%
c	19	0	2	90,4%
d	25	0	2	92,6%
e	23	0	2	92%
f	9	0	3	75%
g	8	0	8	100%
h	12	0	12	100%
Total	7	0	0	91,4% (rata-rata)

Berdasarkan tabel 9 pada gambar a hingga gambar f hasil dari program dengan nilai hitung manual menunjukkan hasil yang berbeda. Kendaraan yang tidak terdeteksi adalah kendaraan yang jaraknya jauh sehingga tidak terlalu terlihat ciri kendaraanya dan kendaraan yang tepat dibawah kamera sehingga tidak terdeteksi sebagai objek yang dikenali pada empat kelas yang ada. Berdasarkan tabel 10, hasil program menunjukkan nilai akurasi (F1) rata-rata untuk pengujian pada citra yang terdiri dari gabungan beberapa kelas kendaraan yang didapat dari persamaan 2 sebesar 91,4%. Dapat disimpulkan bobot dari hasil *training* untuk proses uji kelas ketiga berjalan dengan baik dan sudah memiliki keandalan yang baik

b) Pengujian Berdasarkan Pengaruh Faktor lingkungan  
 Pengujian berdasarkan pengaruh faktor lingkungan dilakukan untuk menguji seberapa akurat bobot dari hasil *training* dalam mendeteksi sebuah kendaraan apabila citra tidak begitu jelas karena faktor lingkungan. Pengujian ini dibagi menjadi tiga bagian yaitu citra ketika cuaca hujan, cuaca berkabut, dan kondisi malam hari. Gambar hasil pengujian pada kondisi hujan dapat dilihat pada gambar 11 di bawah.



Gambar 11. Hasil pengujian citra pada kondisi hujan

Hasil deteksi, klasifikasi, dan perhitungan pada program dan perhitungan manual pada pengujian kondisi hujan dapat dilihat pada tabel 11.

Tabel 11. Perbandingan hasil program dan hitung manual untuk data uji pada kondisi hujan

Gambar	Hasil Program					Nilai Hitung Manual				
	B	C	M	T	Sum	B	C	M	T	Sum
a	0	3	0	0	3	0	3	0	0	3
b	0	11	0	0	11	0	12	0	0	12
c	0	30	0	3	33	0	34	0	4	38
d	0	17	5	0	22	0	19	6	0	25

Parameter TP, FP, dan FN dari gambar a, b, c, dan d dapat dilihat pada tabel 12 berikut

Tabel 12. TP, FP, FN, dan F1 pada data uji kondisi hujan

Gambar	TP	FP	FN	F1
a	3	0	0	100%
b	11	0	1	91,6%
c	33	0	5	86,8%
d	22	0	3	88%
Total	69	0	9	88,4% (rata-rata)

Berdasarkan tabel 11, pada gambar b, c, dan d hasil dari program dengan nilai hitung manual menunjukkan hasil yang berbeda. Perbedaan ini disebabkan karena objek yang dideteksi akibat hujan menjadi kurang jelas. Hujan menyebabkan beberapa objek yang cukup jauh posisinya menjadi kurang jelas sehingga sulit dideteksi. Berdasarkan tabel 12, hasil program menunjukkan nilai akurasi (F1) rata-rata untuk pengujian pada kondisi hujan yang didapat dari persamaan 2 sebesar 88,4%. Dapat disimpulkan bobot dari hasil *training* untuk proses uji pada kondisi hujan masih cukup baik dan memiliki keandalan yang cukup baik meskipun tidak sebaik saat tidak hujan. Untuk gambar hasil pengujian pada kondisi berkabut dapat dilihat pada gambar 12.



Gambar 12. Hasil pengujian citra pada kondisi berkabut

Hasil deteksi, klasifikasi, dan perhitungan pada program dan perhitungan manual pada pengujian kondisi berkabut dapat dilihat pada tabel 13.

Tabel 13. Perbandingan hasil program dan hitung manual untuk data uji pada kondisi berkabut

Gambar	Hasil Program					Nilai Hitung Manual				
	B	C	M	T	Sum	B	C	M	T	Sum
a	0	20	0	0	20	0	27	0	0	27
b	0	2	0	0	2	0	4	0	0	4
c	0	1	0	0	1	0	2	0	0	2
d	0	1	1	0	2	0	1	2	0	3

Parameter TP, FP, dan FN dari gambar a, b, c, dan d dapat dilihat pada tabel 14 berikut.

Tabel 14. TP, FP, FN, dan F1 pada data uji kondisi berkabut

Gambar	TP	FP	FN	F1
a	20	0	7	74%
b	2	0	2	50%
c	1	0	1	50%
d	2	0	1	66,6%
Total	69	0	9	70% (rata-rata)

Berdasarkan tabel 13 pada seluruh gambar hasil dari program dengan nilai hitung manual menunjukkan hasil yang berbeda. Perbedaan ini disebabkan karena objek yang dideteksi akibat kabut menjadi kurang jelas. Kabut menyebabkan beberapa objek yang tertutup kabut dengan intensitas yang cukup tinggi menjadi kurang jelas sehingga sulit dideteksi. Berdasarkan tabel 14, hasil program menunjukkan nilai akurasi (F1) rata-rata untuk pengujian pada kondisi berkabut yang didapat dari persamaan 2 sebesar 70%. Dapat disimpulkan bobot dari hasil *training* untuk proses uji pada cuaca berkabut kurang begitu baik dan memiliki keandalan yang kurang baik dibandingkan saat cuaca cerah ataupun saat hujan. Untuk gambar hasil pengujian pada kondisi malam hari dapat dilihat pada gambar 13 di bawah



Gambar 13. Hasil pengujian citra pada kondisi malam hari

Hasil deteksi, klasifikasi, dan perhitungan pada program dan perhitungan manual pada pengujian kondisi malam hari dapat dilihat pada tabel 15.

Tabel 15. Perbandingan hasil program dan hitung manual untuk data uji pada kondisi malam hari

Gambar	Hasil Program					Nilai Hitung Manual				
	B	C	M	T	Sum	B	C	M	T	Sum
a	0	19	3	0	22	2	25	3	1	31
b	0	1	0	0	1	0	1	0	0	1
c	0	1	1	0	2	0	2	0	0	2
d	0	34	3	0	37	0	40	3	1	44

Parameter TP, FP, dan FN dari gambar a, b, c, dan d dapat dilihat pada tabel 16 berikut.



Tabel 16. TP, FP, FN, dan F1 data uji kondisi malam hari

Gambar	TP	FP	FN	F1
a	22	0	9	70,9%
b	1	0	0	100%
c	1	0	1	50%
d	37	0	7	84,1%
Total	69	0	9	70% (rata-rata)

Berdasarkan tabel 15 pada gambar a, c, dan d hasil dari program dengan nilai hitung manual menunjukkan hasil yang berbeda yang disebabkan karena kurangnya pencahayaan dari objek yang dideteksi dan minimnya data latih pada malam hari. Berdasarkan tabel 16, hasil program menunjukkan nilai akurasi (F1) rata-rata untuk pengujian pada kondisi malam hari sebesar 70%. Dapat disimpulkan bobot dari hasil *training* memiliki keandalan yang kurang begitu baik dibandingkan saat cuaca cerah ataupun saat hujan

### 4.3. Komunikasi Data

#### a) Komunikasi Serial

Komunikasi *Serial* dilakukan agar program deteksi kendaraan dapat dihubungkan dengan mikrokontroler. Program akan mengirim sinyal ke mikrokontroler dan didapat keluaran pada python seperti gambar 7 berikut.

```

960                               1828
Bus : 0                           Bus : 0
Car : 20                          Car : 9
Motorbike : 0                     Motorbike : 0
Truck : 0                          Truck : 0
Number of detected vehicles : 20   Number of detected vehicles : 9
40                                 118
1                                  2
-----

```

Gambar 14. Keluaran pada python pada citra data uji

Pada gambar 14, dapat dilihat bahwa tidak terdapat error ketika program dijalankan. Angka 1 dan 2 adalah angka yang mewakili *serial* pada mikrokontroler. Angka 1 dan 2 artinya python mengirimkan sinyal kepada mikrokontroler untuk mengaktifkan lampu 1 dan 2.

#### b) Komunikasi Database

Komunikasi *database* digunakan agar program deteksi kendaraan pada python dapat dihubungkan dengan file *database*. File *database* yang digunakan adalah csv. Untuk proses membaca *database*, didapat keluaran pada python seperti gambar 14 di atas.

Pada gambar 14, dapat dilihat bahwa tidak terdapat kesalahan ketika program dijalankan. Angka 960 dan 1828 merupakan jumlah kendaraan yang terdapat pada *database* dengan nama file "*base.csv*" yaitu 460 dan 1328 ditambah dengan 500. Untuk proses menulis, jumlah kendaraan akan tersimpan pada *database* dengan nama "*base.csv*".

## 5. Kesimpulan

Pada penelitian ini, didapatkan kesimpulan yaitu pada analisis hasil *training*, didapatkan parameter mAP sebesar 79,11% dan *avg* sebesar 1,91% pada iterasi ke-10.000, pada pengujian citra uji berdasarkan kelas, didapatkan nilai akurasi sebesar 95,7%, pada pengujian citra uji berdasarkan faktor lingkungan, didapatkan nilai akurasi sebesar 78%, bobot hasil *training* lebih handal dalam mendeteksi citra ketika cuaca cerah ataupun siang hari disebabkan faktor cahaya dan tingkat kejelasan objek pada suatu citra, pada pengujian komunikasi data, didapatkan nilai keberhasilan 100% dan seluruh hasil deteksi dan klasifikasi dapat dikomunikasikan sesuai dengan instruksi ke mikrokontroler dan file *database*. Dari hasil Penelitian ini, disarankan untuk dapat dilakukan penelitian lagi untuk melanjutkan atau mengembangkan penelitian ini seperti menambahkan jumlah data latih dan variasinya untuk mendapatkan hasil yang lebih baik dan akurat mengingat CNN bergantung pada jumlah data latih dan variasi data latih yang digunakan.

## Referensi

- [1]. M. H. B. Pratama, A. Hidayatno dan A. A. Zahra "Aplikasi Deteksi Gerak Pada Kamera Keamanan Menggunakan Metode Background Subtraction dengan Algoritma Gaussian Mixture Model," *Transient* vol. 6 no.2. ISSN: 2302-9927
- [2]. R. R. Hirdan dan E. Rainarli, "Perbandingan Metode Run-Length dan Metode Co-Occurance Untuk Ekstraksi Ciri Dengan Metode K-Mean Dalam Mengenal Objek Berdasarkan Tekstur," Diploma Thesis, Program Studi Teknik Informatika, Universitas Komputer Indonesia, 2015.
- [3]. S. R. Dewi dan F. Hakim, "Deep Learning Object Detection Pada Video Menggunakan Tensorflow dan Convolutional Neural Network," Tugas Akhir Program Studi Statistika, Universitas Islam Indonesia, Yogyakarta, 2018
- [4]. K. C. Wang, Y. D. Pranata dan J. C. Wang, "Automatic Vehicle Classification Using Center Strengthened Convolutional Neural Network," dipresentasikan di APSIPA Annual Summit and Conf., Kuala Lumpur, Malaysia, Dec. 12-15, 2017
- [5]. R. Munir, *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*, Bandung: Informatika, 2004.
- [6]. R. K. S. C. Putri, "Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network untuk Klasifikasi Gambar," Tugas Akhir, Program Studi Statistika, Universitas Islam Indonesia, Yogyakarta, 2018.
- [7]. K. O'Shea dan N. Ryan, *An Introduction to Convolutional Neural Network*, ArXiv: 1511.08458, 2015.

- [8]. Arfian dan F. Hakim, "Implementasi Convolutional Neural Network Terhadap Transportasi Tradisional Menggunakan Keras," Tugas Akhir, Program Studi Statistika, Universitas Islam Indonesia, 2018.
- [9]. A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," IBM Journal of Research and Development, vol. 3, pp. 210-229, 1959.
- [10]. L. Deng dan D. Yu, Deep Learning: Methods and Applications, Foundations and Trends® in Signal Processing, vol. 7, no. 3-4, pp. 197-387, 2013.
- [11]. M. Lutz, Learning Python Fifth Edition, Sebastopol: O'Reilly Media, Inc., 2013.
- [12]. J. Redmon, S. Divvala, R. Girshick dan A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," dipresentasikan di CVPR, Las Vegas, Amerika Serikat, Jun. 26 – Jul. 1, 2016.
- [13]. K. A. Shianto, K. Gunadi dan E. Setyati, "Deteksi Jenis Mobil Menggunakan Metode YOLO dan Faster R-CNN," Jurnal Infra, vol. 7 no. 1 pp. 157-163, 2019.
- [14]. I. S. E. Putra, A. Y. Wijaya dan R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101," Jurnal Teknik ITS, vol. 5 no.1, pp. A65-A69 ISSN: 2337-3539.
- [15]. Salsabila. *Penerapan Deep Learning Menggunakan CNN Untuk Klasifikasi Citra Wayang Punakawan*, Tugas Akhir, Jurusan Statistika, Universitas Islam Indonesia