

# Klasifikasi Jenis Kendaraan Darat di Indonesia Menggunakan Pendekatan *Deep Learning*

Muh. Nur Fajrin Amiruddin, Dini Adni Navastara, dan Anny Yuniarti  
Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS)  
*e-mail*: dini\_navastara@if.its.ac.id;

**Abstrak**—Proses klasifikasi jenis kendaraan dengan menggunakan pendekatan *Deep Learning* dalam hal ini menggunakan metode Convolutional Neural Networks (CNN), ada beberapa masalah yang muncul yaitu ketergantungan pada data pelatihan yang besar dan generalisasi data, CNN memerlukan dataset pelatihan yang besar dan bervariasi untuk melakukan pelatihan yang efektif. Jumlah data pelatihan yang terbatas dapat menyebabkan *overfitting*, di mana model menjadi terlalu beradaptasi dengan data pelatihan dan tidak dapat melakukan generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya, sehingga membuat proses klasifikasi jenis kendaraan tidak maksimal. Penelitian ini bertujuan untuk melakukan klasifikasi jenis kendaraan darat di Indonesia ke dalam 5 kelas yaitu Bus, Minibus, Sedan, SUV, dan Truk menggunakan 5 (lima) jenis model arsitektur CNN yaitu ResNet50V2, MobileNetV2, InceptionV3, Xception, dan InceptionResNetV2, dimana dataset yang digunakan bersumber dari internet sebanyak 750 citra kendaraan melalui pengumpulan dataset dengan menggunakan teknik web *scraping*. Hasil eksperimen menunjukkan bahwa model terbaik didapatkan oleh ResNet50V2 pada skenario uji coba K-Fold dengan Augmentasi diperoleh *accuracy* sebesar 0,9906, *precision* sebesar 0,9916, *recall* sebesar 0,9907, dan *F1-score* sebesar 0,9911. Model terbaik diuji cobakan pada data baru berupa rekaman video yang diperoleh dari CCTV Gate Institut Teknologi Sepuluh Nopember (ITS). Hasil akhir akurasi yang didapatkan oleh model ResNet50V2 adalah sebesar 0,65.

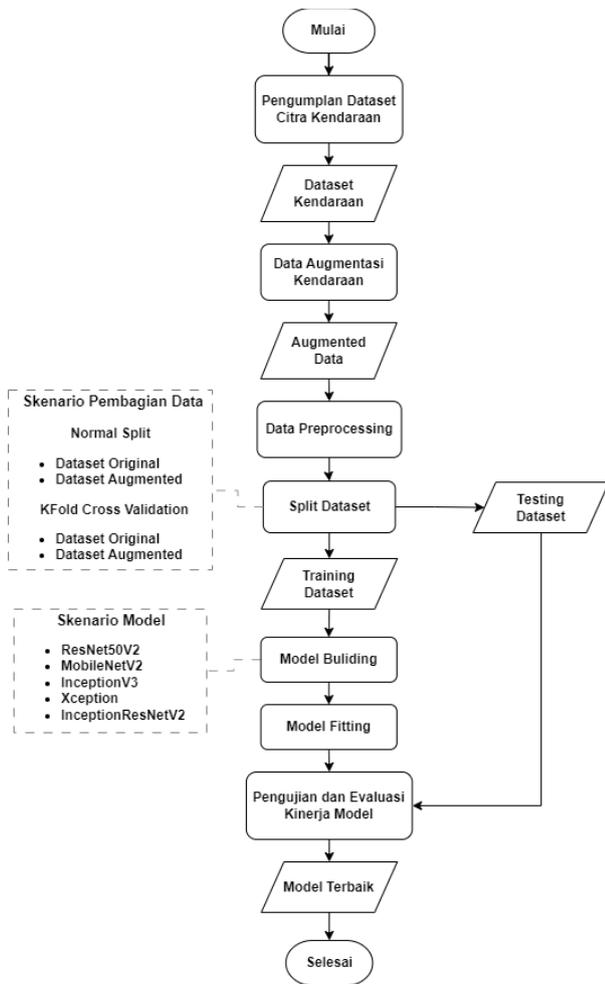
**Kata Kunci**—CNN, *Deep Learning*, Jenis Kendaraan Darat, Klasifikasi, YOLOv3-Tiny.

## I. PENDAHULUAN

DALAM sistem transportasi *modern*, pengelolaan lalu lintas dan pengumpulan data transportasi menjadi hal utama untuk meningkatkan efisiensi, keamanan, dan keberlanjutan. Berdasarkan peningkatan jumlah kendaraan yang terjadi sampai dengan akhir tahun 2022 mencapai 165 juta unit kendaraan atau terjadi peningkatan sebesar 14,73% dari tahun sebelumnya, perlu adanya klasifikasi jenis kendaraan di jalan tol menjadi fokus utama bagi para ahli transportasi dan pengelola infrastruktur jalan [1]. Klasifikasi jenis kendaraan merupakan proses identifikasi dan pengelompokan kendaraan berdasarkan berbagai atribut seperti ukuran, berat, jenis bahan bakar, dan jumlah gandar. Tujuan utama klasifikasi adalah memungkinkan untuk mengelola lalu lintas dengan lebih efisien, menghitung tarif tol berdasarkan golongan dan jenis kendaraan, serta menyediakan data yang akurat untuk perencanaan infrastruktur transportasi yang lebih baik. Jenis kendaraan perlu dilakukan klasifikasi dengan benar karena hasil klasifikasi yang akurat sangat penting untuk keperluan seperti sistem pengawasan lalu lintas, manajemen parkir, dan keamanan jalan raya. Sistem pengawasan digunakan untuk

berbagai tujuan, misalnya pencarian kendaraan dan pemantauan lalu lintas secara *real-time* yang dapat digunakan oleh petugas kepolisian seperti mengidentifikasi kejahatan kendaraan ilegal [2]. Mengumpulkan dan menganalisis data lalu lintas untuk mengoptimalkan penggunaan rute, meningkatkan keamanan transportasi, dan membangun rencana transportasi masa depan [3].

Beberapa penelitian yang telah dilakukan mengenai klasifikasi jenis kendaraan dengan menggunakan metode *Deep Learning* antara lain, penelitian yang dilakukan oleh Suguitan dan Dacaymat menggunakan metode SVM, Logistic Model Trees, Simple Logistic, dan Random Forest. Namun metode yang digunakan dalam penelitian tersebut masih terdapat kelemahan dalam melakukan ekstraksi fitur secara manual dari citra kendaraan, seperti, bentuk, tekstur, warna, dan fitur geometris lainnya [4]. Penelitian selanjutnya yang dilakukan oleh Navastara, Maulana, Suciati, dan Sarena menggunakan metode Hierarchical Multi-SVM dalam melakukan pelatihan model dan klasifikasi kendaraan pada data rekaman CCTV. Hasil uji coba yang didapatkan pada penelitian tersebut dengan *accuracy* sebesar 80,28%, *precision* sebesar 94,27%, *recall* sebesar 73,79%, dan *F-measure* sebesar 82,76% [5]. Pada penelitian yang dilakukan oleh Maungmai dan Nuthong membahas tentang klasifikasi kendaraan menggunakan metode CNN. Penelitian ini melakukan klasifikasi kendaraan dengan dua jenis karakteristik yaitu berdasarkan tipe dan warna. *Dataset* yang digunakan pada penelitian ini adalah sebanyak 914 citra kendaraan dengan pembagian untuk data *training* 75% (686 citra) dan data *testing* 25% (228 citra). Hasil eksperimen yang telah dilakukan menunjukkan bahwa untuk klasifikasi jenis tipe kendaraan model mencapai akurasi 81,62%, sedangkan untuk klasifikasi jenis warna kendaraan model mencapai akurasi 70,09% [2]. Pada penelitian yang dilakukan oleh Roecker, Costa, Almeida, dan Matsushita menggunakan metode CNN untuk klasifikasi jenis kendaraan dengan menggunakan gambar beresolusi rendah dari perspektif kamera depan. Hasil eksperimen pada *subset* Beijing Institut of Technology (BIT)-Vehicle Dataset dengan sampel yang terdistribusi secara merata sebanyak 476 data untuk masing-masing kelas yang terdiri dari Bus, Microbus, Minivan, Sedan, SUV, dan Truck. Hasil eksperimen menunjukkan bahwa model mencapai akurasi 93,90% [6]. Penelitian selanjutnya yang dilakukan oleh Hicham, Ahmed, dan Mohammed tentang klasifikasi tipe kendaraan menggunakan metode CNN. *Dataset* yang digunakan pada eksperimen ini sebanyak 2400 sampel data yang terbagi menjadi 4 bagian kategori, yaitu, School Bus, Ambulance, Police, dan Moroccan Transport Company (CTM). Hasil akurasi model yang didapatkan pada eksperimen ini adalah 89% [7]. Penelitian tentang klasifikasi jenis kendaraan menggunakan



Gambar 1. Tahapan Proses Penelitian.

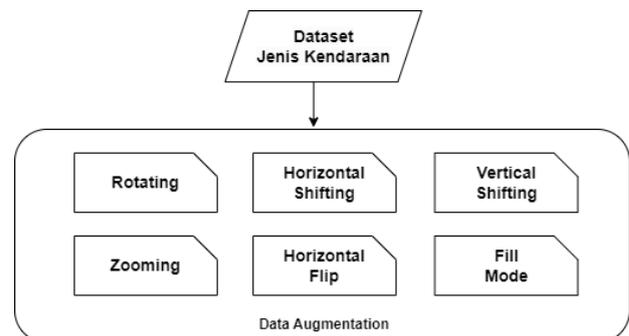
metode CNN yang dilakukan oleh Fadlia dan Kosasih, *dataset* yang digunakan dalam penelitian tersebut sebanyak 120 citra yang terdiri dari citra mobil, motor dan sepeda. Hasil uji coba dan evaluasi model terhadap tiga jenis kendaraan menggunakan *package* Keras menunjukkan akurasi sebesar 94,4% pada tahap pelatihan dan 73,3% pada tahap pengujian [8]. Pada penelitian yang dilakukan oleh Bhujbal dan Mane tentang klasifikasi tipe kendaraan menggunakan metode YOLO. Penelitian yang dilakukan mengenai pendeteksian tipe kendaraan berdasarkan setiap objek pada model yang dilatih. Pada eksperimen ini menggunakan *dataset* yang diambil dari *cityscapes* sebanyak 2659 sampel data *training* dan 500 sampel data *testing*. Setiap *dataset* terdistribusi pada masing-masing kelas Car, Bus, Truck, Motorbike dengan akurasi model yang didapatkan untuk masing-masing kelas secara berurutan adalah 93,05%, 85,10%, 87,83%, 82,02%. Berdasarkan pengujian menggunakan model YOLO rata-rata akurasi yang didapatkan untuk eksperimen ini mencapai 87% [9]. Dalam melakukan deteksi pengenalan ekspresi wajah manusia dengan menggunakan metode YOLO berbasis CNN yang dilakukan oleh Nafis, Navastara, dan Yuniarti terdapat tujuh kelas ekspresi wajah yang dapat dikenali, yaitu marah, jijik, takut, senang, sedih, terkejut, dan netral. Berdasarkan hasil eksperimen, akurasi terbaik dari *dataset* gambar diam adalah 94% pada *dataset* CK+ dengan tiga *channel* dan *learning rate* 0.01. Selain itu, akurasi data video dengan berbagai pose wajah mencapai 73% [10]. Penelitian yang dilakukan oleh Padilla, Netto, dan da Silva membahas tentang metrik kinerja



Gambar 2. Sampel *Dataset* Citra Kendaraan.

Tabel 1. Rincian Jumlah *Dataset* Citra Kendaraan

No	Jenis Kendaraan	Jumlah Data
1	Bus	23
2	Minibus	33
3	Sedan	32
4	SUV	33
5	Truk	29
<b>Total Dataset</b>		<b>150</b>



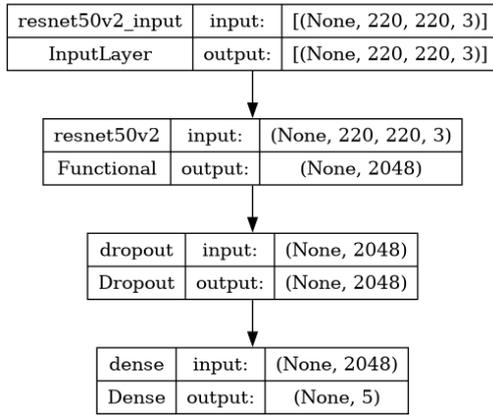
Gambar 3. Detail Data Augmentasi Kendaraan.

yang paling populer yang digunakan untuk mengevaluasi algoritma deteksi objek, termasuk konsep utama dari penelitian dan perbandingan antara implementasi yang berbeda dapat menghasilkan hasil akhir yang berbeda [11]. Dalam beberapa kondisi, terdapat informasi yang lebih rinci dan berisi parameter transformasi *linear* atau *non-linear*. Sistem deteksi objek membangun model untuk kelas objek terdiri dari serangkaian contoh pelatihan. Dalam kasus *fixed object* hanya satu contoh yang diperlukan, tetapi lebih umum beberapa contoh pelatihan yang diperlukan untuk menangkap aspek-aspek tertentu dari variabilitas kelas [12].

Dalam buku Suyanto mengenai penerapan metode Augmentasi dan K-Fold Cross Validation dapat digunakan untuk mengatasi kelemahan pada metode CNN. Augmentasi dapat digunakan untuk mengatasi data pelatihan yang terbatas sehingga tidak terjadi *overfitting* dengan meningkatkan jumlah dan variasi *dataset* untuk melakukan transformasi pada gambar, seperti *flip*, *random cropping*, *color jitter* dan *random combination*. Sedangkan penerapan metode K-Fold Cross Validation membantu meminimalkan bias dalam evaluasi kinerja dan memberikan perkiraan kinerja yang lebih

Tabel 2.  
Pembagian Data *Train* dan Data *Test*

Kelas	Dataset Asli Tanpa Augmentasi		Dataset Asli Dengan Augmentasi	
	Data <i>Train</i>	Data <i>Test</i>	Data <i>Train</i>	Data <i>Test</i>
Bus	127	23	340	110
Minibus	117	33	363	87
Sedan	118	32	359	91
SUV	117	33	365	85
Truk	121	29	373	77
<b>Total</b>	<b>600</b>	<b>150</b>	<b>1800</b>	<b>450</b>



Gambar 4. Visualisasi Arsitektur Model ResNet50V2.

stabil dan andal. Metode ini juga memastikan bahwa model dapat digeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya [13]. Metode K-Fold Cross Validation sangat berguna ketika *dataset* relatif kecil dan tidak ada *dataset* validasi yang terpisah, dengan membagi data menjadi beberapa *fold*, sehingga dapat menggunakan semua data untuk pelatihan dan evaluasi. Dimana metode ini terbukti dapat meningkatkan akurasi hasil klasifikasi [14-15].

Pada penelitian ini akan digunakan metode CNN dalam melakukan klasifikasi jenis kendaraan darat, kemudian akan menerapkan data augmentasi pada citra kendaraan dan K-Fold Cross Validation.

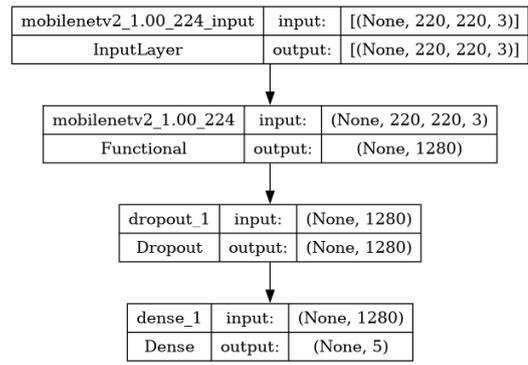
## II. METODOLOGI

### A. Tahapan Penelitian

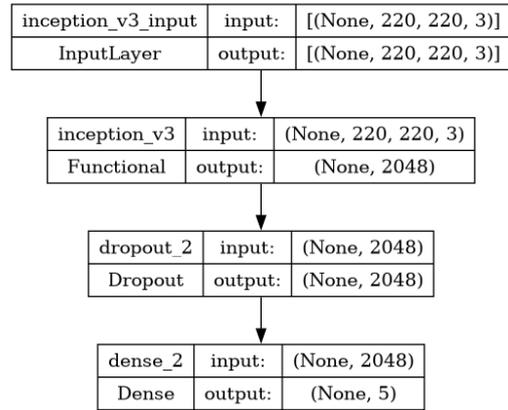
Tahapan penelitian yang digunakan dalam melakukan klasifikasi jenis kendaraan darat ditunjukkan pada Gambar 1. Tahapan proses pada penelitian ini antara lain, melakukan pengumpulan *dataset* citra kendaraan, data augmentasi kendaraan, data *preprocessing*, pembagian data *train* dan data *test*, pemodelan *deep learning*, serta pengujian dan evaluasi kinerja model.

### B. Pengumpulan Dataset

*Dataset* citra jenis kendaraan dikumpulkan dari berbagai sumber di internet dengan menggunakan teknik *Web Scraping*, kemudian memasukkan kata kunci jenis kendaraan yang akan di *download*, selanjutnya dilakukan seleksi citra kendaraan yang akan digunakan sebagai *dataset*. *Dataset* terdiri dari 5 (lima) jenis kendaraan yaitu Bus, Minibus, Sedan, SUV, dan Truk. Adapun sampel *dataset* citra pada masing-masing jenis kendaraan ditunjukkan pada Gambar 2.



Gambar 5. Visualisasi Arsitektur Model MobileNetV2.



Gambar 6. Visualisasi Arsitektur Model InceptionV3.

Rincian *dataset* jenis kendaraan yang diperoleh dari hasil *scraping image* diuraikan pada Tabel 1.

### C. Data Augmentasi Kendaraan

Data augmentasi adalah teknik dalam pengolahan data yang digunakan dalam *machine learning* untuk meningkatkan kualitas dan jumlah data pelatihan dengan membuat variasi data.

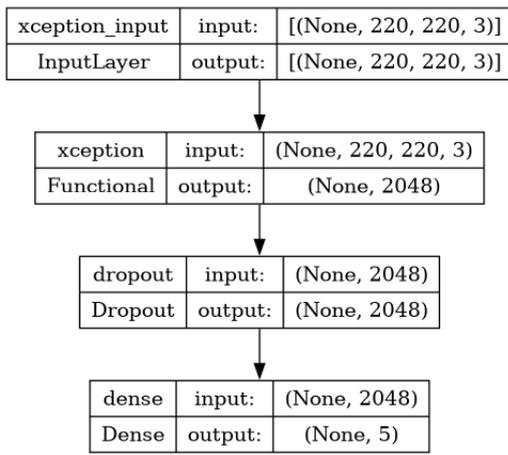
Tujuan dari data augmentasi adalah memberikan lebih banyak variasi pada *dataset* pelatihan, sehingga model yang dibangun dapat lebih baik dalam mengatasi berbagai kondisi dan variasi yang mungkin muncul pada data uji. Jenis augmentasi yang dilakukan terdiri dari *rotating*, *horizontal shifting*, *vertical shifting*, *zooming*, *horizontal flip*, dan *fill mode* seperti ditunjukkan pada Gambar 3.

### D. Data Preprocessing

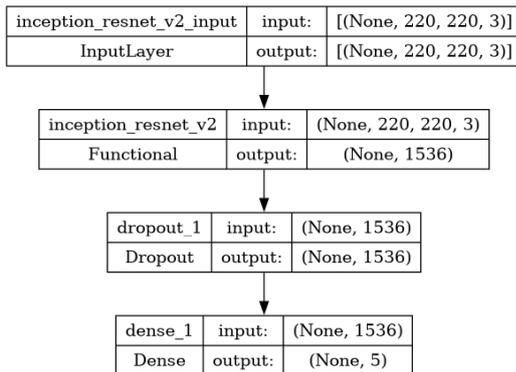
*Dataset* yang sudah diberikan kode label untuk masing-masing jenis kendaraan, sebelum dilakukan pemodelan metode maka terlebih dahulu dilakukan *preprocessing* seperti mengatur ukuran *pixel* citra ke dalam satu ukuran citra yang sama, kemudian dilakukan proses normalisasi citra dan label *encoding*. Proses label *encoding* menggunakan fungsi *LabelEncoder* untuk label jenis kendaraan dalam bentuk *array/list* agar bisa diproses pada tahap pemodelan selanjutnya.

### E. Pembagian Data Train dan Data Test

*Dataset* yang digunakan untuk pelatihan model dibagi menjadi 2 (dua) bagian. *Dataset* asli tanpa augmentasi yang berjumlah 750 citra kemudian dibagi menjadi data *training* dan data *testing*. Kemudian menggunakan *dataset* asli dengan augmentasi yang berjumlah 2.250 citra. *Dataset* dibagi



Gambar 7. Visualisasi Arsitektur Model Xception.



Gambar 8. Visualisasi Arsitektur Model InceptionResNetV2

menjadi 80% untuk data *training* dan 20% untuk data *testing*. Pembagian dataset ditunjukkan pada Tabel 2 untuk distribusi jumlah data *train* dan *test* pada setiap kelas.

F. K-Fold Cross Validation

Pada penelitian ini menggunakan penggabungan *dataset* (*dataset* asli dan *dataset* hasil augmentasi) berjumlah 2.250 data yang sudah diolah akan dibagi menjadi 10 (*fold*) bagian secara bergantian, setiap bagian secara berurutan dijadikan sebagai data *testing*, sedangkan bagian lainnya sebagai data *training*. Oleh karena itu, semua *dataset* akan menjadi data *train* dan data *test*.

G. Pemodelan Deep Learning

Pemodelan dilakukan dengan mendefinisikan arsitektur model *pre-trained* yang digunakan yaitu ResNet50V2, MobileNetV2, InceptionV3, Xception, dan InceptionResnetV2. Proses *training* model dilakukan dengan memasukkan data *train* ke dalam setiap model dan menggunakan parameter yang sama selama pelatihan.

Tahapan pelatihan yang dilakukan seperti ditunjukkan pada Gambar 4, Gambar 5, Gambar 6, Gambar 7, dan Gambar 8 terdiri dari beberapa *layer* diantaranya *input layer*, *functional*, *dropout*, dan *dense*. *Input layer* berisikan citra yang akan diklasifikasi dengan ukuran 220x220 piksel dan 3 *channel* yang mewakili informasi warna. *Functional* memiliki beberapa *layer* yang berfungsi sebagai operasi konvolusi pada *input* citra dengan *filter* untuk mengekstrak fitur penting. *Dropout layer* digunakan untuk mencegah terjadinya *overfitting* saat pelatihan. *Dense layer* digunakan untuk memetakan fitur yang diekstrak ke dalam 5 kelas klasifikasi.

Tabel 3. Rangkuman Skenario Pemodelan.

Dataset	Pembagian Data	Variabel Konstan
Original	Split Normal	<ul style="list-style-type: none"> <li>Variasi Model <i>Pre-trained</i>:                             <ol style="list-style-type: none"> <li>ResNet50V2</li> <li>MobileNetV2</li> <li>InceptionV3</li> <li>Xception</li> <li>InceptionResnetV2</li> </ol> </li> <li>Batch Size: 32</li> <li>Image Size: 220 x 220 pixel</li> <li>Loss Function: Categorical Crossentropy</li> <li>Optimizer: Adam</li> </ul>
Original + Augmentasi	Split Normal	
Original	K Fold	
Original + Augmentasi	K Fold	

Tabel 4. Detail Jumlah Citra Tanpa Augmentasi

Kelas	Pembagian Data		
	Train	Test	Jumlah
Bus	127	23	150
Minibus	117	33	150
Sedan	118	32	150
SUV	117	33	150
Truk	121	29	150
<b>Total</b>	<b>600</b>	<b>150</b>	<b>750</b>

1) ResNet50V2

Tahapan visualisasi arsitektur model *pre-trained* ResNet50V2 yang digunakan saat pelatihan model ditunjukkan pada Gambar 4.

2) MobileNetV2

Tahapan visualisasi arsitektur model *pre-trained* MobileNetV2 yang digunakan saat pelatihan model ditunjukkan pada Gambar 5.

3) InceptionV3

Tahapan visualisasi arsitektur model *pre-trained* InceptionV3 yang digunakan saat pelatihan model ditunjukkan pada Gambar 6.

4) Xception

Tahapan visualisasi arsitektur model *pre-trained* Xception yang digunakan saat pelatihan model ditunjukkan pada Gambar 7.

5) InceptionResnetV2

Tahapan visualisasi arsitektur model *pre-trained* InceptionResNetV2 yang digunakan saat pelatihan model ditunjukkan pada Gambar 8.

Arsitektur yang digunakan merupakan modifikasi dari model *pre-trained* dengan beberapa perubahan yang dilakukan pada *layer* akhir.

H. Pengujian Model dan Evaluasi Kinerja Model

Pada proses pengujian model, setelah melakukan pelatihan model maka untuk memastikan kualitas kinerja dan keandalan dari model yang didapatkan akan dilakukan pengujian model menggunakan data *test*. Dari pengujian tersebut akan diketahui kesesuaian dari model untuk melakukan prediksi atau klasifikasi citra dan label jenis

Tabel 3.  
Kinerja Model Skenario Split Normal Tanpa Augmentasi

Model	Precision	Recall	F1-Score	Accuracy
ResNet50V2	0,9515	0,9508	0,9510	0,9467
MobileNetV2	0,8876	0,8867	0,8869	0,8800
InceptionV3	0,8879	0,8857	0,8856	0,8800
Xception	0,8785	0,8732	0,8755	0,8667
InceptionResnetV2	0,8990	0,8951	0,8967	0,8933

Tabel 4.  
Detail Jumlah Citra Dengan Augmentasi

Kelas	Pembagian Data		
	Train	Test	Jumlah
Bus	340	110	450
Minibus	363	87	450
Sedan	359	91	450
SUV	365	85	450
Truk	373	77	450
<b>Total</b>	<b>1800</b>	<b>450</b>	<b>2250</b>

kendaraan kemudian akan dibandingkan dengan aktual label jenis kendaraan.

Evaluasi kinerja model CNN yang dihasilkan untuk setiap jenis model dilakukan pengujian dan evaluasi model dengan menggunakan beberapa metrik evaluasi yang ditunjukkan pada Persamaan (1) *Accuracy*, Persamaan (2) *Precision*, Persamaan (3) *Recall*, dan Persamaan (4) *F1-score*.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (1)$$

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

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

$$F1-Score = \frac{(2 \times Precision \times Recall)}{(Precision+Recall)} \quad (4)$$

*True Positive* (TP) merupakan jumlah sampel positif yang diklasifikasikan dengan benar ke dalam kelas positif oleh model. *True Negative* (TN) merupakan jumlah sampel negatif yang diklasifikasikan dengan benar ke dalam kelas negatif oleh model. *False Positive* (FP) merupakan jumlah sampel negatif yang salah diklasifikasikan sebagai positif oleh model. *False Negative* (FN) merupakan jumlah sampel positif yang salah diklasifikasikan sebagai negatif.

### III. HASIL DAN PEMBAHASAN

#### A. Skenario Pemodelan

Skenario pemodelan dilakukan sebanyak 4 (empat) kali untuk mendapatkan skenario yang terbaik untuk membuat model klasifikasi jenis kendaraan dengan membandingkan performa model yang didapatkan. Adapun skenario pemodelan yang dilakukan untuk setiap model *pre-trained* yang digunakan ditunjukkan pada Tabel 3.

#### B. Skenario Split Normal Tanpa Augmentasi

Skenario pertama menggunakan *dataset* original tanpa augmentasi yang diperoleh dari *scraping image*.

*Dataset* terdiri dari 750 citra kendaraan yang terbagi atas 5 kelas, sehingga pembagian untuk masing-masing kelas

Tabel 1.  
Kinerja Model Skenario Split Normal Dengan Augmentasi

Model	Precision	Recall	F1-Score	Accuracy
ResNet50V2	0,98	0,98	0,98	0,9822
MobileNetV2	0,97	0,97	0,97	0,9689
InceptionV3	0,96	0,96	0,96	0,9622
Xception	0,95	0,95	0,95	0,9489
InceptionResnetV2	0,95	0,95	0,95	0,9511

Tabel 2.  
Detail Jumlah Citra K-Fold Tanpa Augmentasi

Kelas	Pembagian Data		
	Train	Test	Jumlah
Bus	135	15	150
Minibus	135	15	150
Sedan	135	15	150
SUV	135	15	150
Truk	135	15	150
<b>Total</b>	<b>675</b>	<b>75</b>	<b>750</b>

adalah 150 citra. Pembagian *dataset* yang dilatih pada pembuatan model menggunakan *split* normal diantaranya data *train* 80% atau 600 citra dan data *test* 20% atau 150 citra ditunjukkan pada Tabel 4.

Berdasarkan pengukuran kinerja model *pre-trained* (ResNet50V2, MobileNetV2, InceptionV3, Xception, dan InceptionResnetV2) pada skenario pertama *split* normal tanpa augmentasi didapatkan hasil kinerja masing-masing model yang ditunjukkan pada Tabel 5.

Berdasarkan Tabel 5 untuk skenario pertama ditunjukkan bahwa model yang memiliki nilai *Accuracy* tertinggi 0,9467 adalah model ResNet50V2 begitu juga untuk nilai *Precision* 0,9515, nilai *Recall* 0,9508, dan nilai *F1-Score* 0,9510.

#### C. Skenario Split Normal Dengan Augmentasi

Dalam skenario kedua menggunakan *dataset* original dengan augmentasi yang diperoleh dari *scraping image*. *Dataset* terdiri dari 2.250 citra kendaraan yang terbagi atas 5 kelas, sehingga pembagian untuk masing-masing kelas adalah 450 citra.

Pembagian *dataset* yang dilatih pada pembuatan model menggunakan *split* normal diantaranya data *train* 80% atau 1.800 citra dan data *test* 20% atau 450 citra ditunjukkan pada Tabel 6.

Berdasarkan pengukuran kinerja model *pre-trained* (ResNet50V2, MobileNetV2, InceptionV3, Xception, dan InceptionResnetV2) pada skenario kedua *split* normal dengan augmentasi didapatkan hasil kinerja masing-masing model yang ditunjukkan pada Tabel 7.

Berdasarkan Tabel 7 untuk skenario kedua ditunjukkan bahwa model yang memiliki nilai *Accuracy* tertinggi 0,9822 adalah model ResNet50V2 begitu juga untuk nilai *Precision* 0,98, nilai *Recall* 0,98, dan nilai *F1-Score* 0,98.

#### D. Skenario K-Fold Tanpa Augmentasi

Dalam skenario ketiga menggunakan *dataset* original tanpa augmentasi yang diperoleh dari *scraping image* dan dilakukan pembagian *dataset* dengan cara K-Fold. *Dataset* terdiri dari 750 citra kendaraan yang terbagi atas 5 kelas, sehingga pembagian untuk masing-masing kelas adalah 150

Tabel 7.  
Kinerja Model Skenario K-Fold Tanpa Augmentasi

Model	Precision	Recall	F1-Score	Accuracy
ResNet50V2	0,97	0,97	0,97	0,9706
MobileNetV2	0,95	0,95	0,95	0,9520
InceptionV3	0,96	0,96	0,96	0,9587
Xception	0,96	0,96	0,96	0,9587
InceptionResnetV2	0,96	0,96	0,96	0,9493

Tabel 8.  
Jumlah Citra K-Fold Dengan Augmentasi

Kelas	Pembagian Data		
	Train	Test	Jumlah
Bus	405	45	450
Minibus	405	45	450
Sedan	405	45	450
SUV	405	45	450
Truk	405	45	450
<b>Total</b>	<b>2025</b>	<b>225</b>	<b>2250</b>



Gambar 1 Pengujian Deteksi Objek Pada Data Video CCTV

citra. Pembagian *dataset* yang dilatih pada pembuatan model menggunakan K-Fold ditunjukkan pada Tabel 8 untuk *split* 1 pada *fold* pertama dimana data *train* sebanyak 675 citra dan data *test* sebanyak 75 citra.

Berdasarkan pengukuran kinerja model *pre-trained* (ResNet50V2, MobileNetV2, InceptionV3, Xception, dan InceptionResnetV2) pada skenario ketiga K-Fold tanpa augmentasi didapatkan hasil kinerja masing-masing model. Berdasarkan Tabel 9 pada skenario ketiga ditunjukkan bahwa model yang memiliki nilai *Accuracy* tertinggi 0,9706 adalah model ResNet50V2.

#### E. Skenario K-Fold Dengan Augmentasi

Dalam skenario keempat menggunakan *dataset* original dengan augmentasi yang diperoleh dari *scraping image* dan dilakukan pembagian *dataset* dengan cara K-Fold. *Dataset* terdiri dari 2250 citra kendaraan yang terbagi atas 5 kelas, sehingga pembagian untuk masing-masing kelas adalah 450 citra. Pembagian *dataset* yang dilatih pada pembuatan model menggunakan K-Fold ditunjukkan pada Tabel 10 untuk *split* 1 pada *fold* pertama dimana data *train* sebanyak 2.025 citra dan data *test* sebanyak 225 citra.

Berdasarkan pengukuran kinerja model *pre-trained* (ResNet50V2, MobileNetV2, InceptionV3, Xception, dan InceptionResnetV2) pada skenario keempat K-Fold dengan augmentasi didapatkan hasil kinerja masing-masing model yang ditunjukkan pada Tabel 11.

Tabel 5.  
Kinerja Model Uji Coba K-Fold Dengan Augmentasi

Model	Precision	Recall	F1-Score	Accuracy
ResNet50V2	0,99	0,99	0,99	0,9906
MobileNetV2	0,98	0,98	0,98	0,9848
InceptionV3	0,97	0,97	0,97	0,9702
Xception	0,97	0,97	0,97	0,9733
InceptionResnetV2	0,97	0,97	0,97	0,9697

Tabel 6.  
Rekap Hasil Model Terbaik Dari Setiap Skenario Pengujian

Skenario Uji Coba	Model	Precision	Recall	F1-Score	Accuracy
Kesatu	ResNet50V2	0,9514	0,9507	0,9510	0,9467
Kedua	ResNet50V2	0,9829	0,9821	0,9822	0,9822
Ketiga	ResNet50V2	0,9712	0,9707	0,9707	0,9706
<b>Keempat</b>	<b>ResNet50V2</b>	<b>0,9916</b>	<b>0,9907</b>	<b>0,9911</b>	<b>0,9906</b>

Tabel 13.  
Classification Report Deteksi Objek CCTV Gate ITS

	Precision	Recall	F1-Score
Bus	0	0	0
Minibus	0,3454	0,9500	0,5067
Sedan	0,2250	0,4500	0,3000
SUV	0,9747	0,6695	0,7938
Truk	1,000	0,3000	0,4615
<b>Akurasi</b>		<b>0,65</b>	

Berdasarkan Tabel 11 pada skenario keempat ditunjukkan bahwa model yang memiliki nilai *Accuracy* tertinggi 0,9906 adalah model ResNet50V2 begitu juga untuk nilai *Precision* 0,99, nilai *Recall* 0,99, dan nilai *F1-Score* 0,99.

#### F. Rekomendasi Model

Setelah dilakukan 4 skenario pemodelan yaitu Split Normal tanpa Augmentasi, Split Normal dengan Augmentasi, K-Fold Cross Validation tanpa Augmentasi, dan K-Fold Cross Validation dengan Augmentasi masing-masing model terbaik dipilih 1 untuk setiap skenario sebagaimana ditunjukkan pada Tabel 12.

Berdasarkan Tabel 12 maka direkomendasikan model klasifikasi yang akan digunakan dalam melakukan klasifikasi maupun deteksi objek jenis kendaraan adalah model ResNet50V2 pada skenario keempat yaitu K-Fold dengan Augmentasi diperoleh nilai *Accuracy* 0,9906, *Precision* 0,9916, *Recall* 0,9907, dan *F1-Score* 0,9911.

#### G. Pengujian Pada Data Video CCTV

Setelah mendapatkan model terbaik yaitu skenario keempat K-Fold dengan augmentasi, kemudian model tersebut dilakukan pengujian menggunakan data baru berupa rekaman video yang diperoleh dari CCTV Gate ITS. Dalam studi kasus ini metode YOLOv3-Tiny digunakan untuk mendeteksi objek dan model *classifier* terbaik sebagai klasifikasi dari jenis kendaraan yang dideteksi. Pengujian dilakukan untuk mengukur seberapa baik kinerja model terbaik dalam hal ini ResNet50V2 dalam melakukan klasifikasi jenis kendaraan. Pengujian deteksi objek pada rekaman CCTV Gate ITS dibagi dalam 3 video yang masing-

masing video berisi 10 objek kendaraan dalam 100 *frame* seperti yang ditunjukkan pada Gambar 9.

Berdasarkan hasil uji coba deteksi objek seperti ditunjukkan pada Tabel 13 menunjukkan bahwa kinerja model klasifikasi yang didapatkan pada skenario keempat belum mampu mendapatkan hasil akurasi yang baik dalam melakukan klasifikasi atau deteksi objek pada rekaman CCTV, hanya didapatkan nilai akurasi sebesar 0,65. Masih terdapat kesalahan klasifikasi pada kelas Sedan dan SUV karena kedua jenis mobil tersebut memiliki bentuk dan ukuran yang hampir sama. Tentunya hal ini perlu dilakukan penelitian lebih lanjut agar bisa didapatkan hasil akurasi yang lebih baik lagi.

#### IV. KESIMPULAN DAN SARAN

Pada penelitian ini klasifikasi jenis kendaraan darat menggunakan metode *deep learning* dilakukan pengumpulan *dataset* dengan menggunakan teknik *web scraping*. Citra yang digunakan pada penelitian ini terdiri dari 2 jenis, yaitu citra tanpa data augmentasi sebanyak 750 citra dan citra dengan data augmentasi sebanyak 2.250 citra. Berdasarkan hasil pelatihan dengan menggunakan data augmentasi berpengaruh secara signifikan diperoleh tingkat akurasi model ResNet50v2 menggunakan K-Fold Cross Validation sebesar 0,9906, *precision* sebesar 0,9916, *recall* sebesar 0,9907, dan *F1-score* sebesar 0,9911.

Pada penelitian selanjutnya untuk mendapatkan hasil klasifikasi yang lebih baik antara kelas Sedan dan SUV dapat melakukan analisis dan optimalisasi fitur lebih lanjut dalam mengatasi hal tersebut. Selain itu, untuk mendapatkan hasil akurasi yang lebih baik dalam mendeteksi objek pada data berupa video dapat menggunakan model deteksi objek yang lebih baik seperti YOLOv5 atau YOLOv8.

#### REFERENSI

- [1] Badan Pusat Statistik, *Statistik Indonesia 2023*. Jakarta: Badan Pusat Statistik (BPS), 2023.
- [2] W. Maungmai and C. Nuthong, "Vehicle Classification With Deep Learning," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, Singapore: IEEE, 2019, pp. 294–298. doi: 10.1109/CCOMS.2019.8821689.
- [3] D. Sharma and Z. A. Jaffery, "Categorical vehicle classification and tracking using deep neural networks," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 9, pp. 564–574, 2021, doi: 10.14569/IJACSA.2021.0120964.
- [4] A. S. Suguitan and L. N. Dacaymat, "Vehicle Image Classification Using Data Mining Techniques," in *Proceedings of the 2nd International Conference on Computer Science and Software Engineering*, New York, NY, USA: ACM, 2019, pp. 13–17. doi: 10.1145/3339363.3339366.
- [5] D. A. Navastara, M. F. Maulana, N. Suciati, and S. T. Sarena, "Vehicle Classification Based on CCTV Video Recording Using Histogram of Oriented Gradients, Local Binary Patterns, and Hierarchical Multi-SVM," in *IOP Conference Series: Materials Science and Engineering, The 5th International Conference on Information Technology and Digital Applications (ICITDA 2020)*, Yogyakarta: IOP Science, 2021. doi: 10.1088/1757-899X/1077/1/012068.
- [6] M. N. Roecker, Y. M. G. Costa, J. L. R. Almeida, and G. H. G. Matsushita, "Automatic Vehicle Type Classification With Convolutional Neural Networks," in *2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Maribor, Slovenia: IEEE, 2018, pp. 1–5. doi: 10.1109/IWSSIP.2018.8439406.
- [7] B. Hicham, A. Ahmed, and M. Mohammed, "Vehicle Type Classification Using Convolutional Neural Network," in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, Marrakech, Morocco: IEEE, 2018, pp. 313–316. doi: 10.1109/CIST.2018.8596500.
- [8] N. Fadlia and R. Kosasih, "Klasifikasi jenis kendaraan menggunakan metode Convolutional Neural Network (CNN)," *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 24, no. 3, pp. 207–215, 2019, doi: 10.35760/tr.2019.v24i3.2397.
- [9] A. Bhujbal and D. T. Mane, "Vehicle Type Classification Using Deep Learning," in *Soft Computing and Signal Processing. ICSCSP 2019. Advances in Intelligent Systems and Computing*, vol. 1118, V. S. Reddy, V. K. Prasad, J. Wang, and K. T. V. Reddy, Eds., Singapore: Springer, 2020, pp. 279–290. doi: 10.1007/978-981-15-2475-2\_26.
- [10] A. F. Nafis, D. Adni Navastara, and A. Yuniarti, "Facial Expression Recognition On Video Data With Various Face Poses Using Deep Learning," in *2020 12th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Yogyakarta, Indonesia: IEEE, 2020, pp. 362–367. doi: 10.1109/ICITEE49829.2020.9271740.
- [11] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A Survey On Performance Metrics For Object-Detection Algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Niteroi, Brazil: IEEE, 2020, pp. 237–242. doi: 10.1109/IWSSIP48289.2020.9145130.
- [12] Y. Amit, P. Felzenszwalb, and R. Girshick, "Object Detection," in *Computer Vision*, Cham, Swiss: Springer International Publishing, 2020, pp. 1–9. doi: 10.1007/978-3-030-03243-2\_660-1.
- [13] S. Suyanto, *Data Mining Untuk Klasifikasi dan Klusterisasi Data*. Bandung: INFORMATIKA, 2019, ISBN: 978-602-6232-36-6.
- [14] V. R. R. Chirra, S. R. Uyyala, and V. K. K. Kolli, "Virtual facial expression recognition using deep CNN with ensemble learning," *J Ambient Intell Humaniz Comput*, vol. 12, no. 12, pp. 10581–10599, 2021, doi: 10.1007/s12652-020-02866-3.
- [15] N. D. Indira and M. V. G. Rao, "Deep learning CNN-Based Hybrid Extreme Learning Machine With Bagging Classifier For Automatic Modulation Classification," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 2s, pp. 134–141, 2022.