

# Sistem Pembatasan Kecepatan *Autonomous Car* di Indonesia

Abdul Azzam Ajhari<sup>1)</sup>

(1) Badan Siber dan Sandi Negara, [abdul.azzam@bssn.go.id](mailto:abdul.azzam@bssn.go.id)

## Abstrak

Perkembangan teknologi mobil yang semakin cepat membutuhkan sistem pengawasan yang lebih canggih dengan artificial intelligence (AI). Kecepatan mobil otomatis perlu dibatasi berdasarkan kondisi cuaca yang terdeteksi oleh kamera dengan penerapan pendekatan AI yang lebih dalam, yaitu deep learning (DL). Pada penelitian ini, menggunakan metode eksperimen dengan teknologi pengolahan gambar yang terintegrasi dengan AI untuk pendeteksian kondisi cuaca saat mobil bergerak. Keunggulan dari pendekatan DL adalah kemampuannya dalam mempelajari pola dan fitur yang lebih kompleks dalam gambar, sehingga mampu mengenali perubahan cuaca dengan lebih baik. Dalam penelitian ini, dilakukan perbandingan kinerja antara algoritma Convolutional Neural Network (CNN) dan ResNet18 dalam mengklasifikasikan gambar berdasarkan kondisi cuaca yang terdeteksi. Hasilnya menunjukkan bahwa kedua algoritma tersebut memiliki keunggulan dan kelemahan masing-masing. CNN memiliki keunggulan dalam mengenali fitur-fitur umum dalam gambar dengan akurasi pelatihan sebesar 84.80% dan akurasi pengujian sebesar 84.00%, sedangkan ResNet18 memiliki kemampuan yang lebih baik dalam mengatasi masalah klasifikasi gambar yang kompleks dan serangan siber pada sistem dari perubahan pendeteksian gambar secara tiba-tiba dengan akurasi pelatihan sebesar 94.80% dan akurasi pengujian sebesar 98.00%. Kontribusi penelitian ini adalah mendeteksi kondisi cuaca secara akurat dengan memberikan informasi yang relevan kepada pengemudi atau sistem pengendalian otomatis, sehingga meningkatkan keamanan siber dan kenyamanan dalam berkendara.

Kata kunci: *autonomous car*, CNN, pendeteksian kondisi cuaca, ResNet18, sistem keamanan siber pengawasan

## Abstract

The rapid development of car technology requires a more sophisticated surveillance system with artificial intelligence (AI). Automatic car speed needs to be limited based on weather conditions detected by the camera by applying a deeper AI approach, namely deep learning (DL). In this study, an experimental method was used with image processing technology integrated with AI to detect weather conditions when the car is moving. The advantage of the DL approach is its ability to learn more complex patterns and features in images, so that it can better recognize changes in the weather. In this study, a performance comparison was made between the Convolutional Neural Network (CNN) algorithm and ResNet18 in classifying images based on detected weather conditions. The results show that both algorithms have their own advantages and disadvantages. CNN has an advantage in recognizing common features in images with a training accuracy of 84.80% and a testing accuracy of 84.00%, while ResNet18 has a better ability to solve complex image classification problems and cyber attacks on the system from sudden changes in image detection. arrived with a training accuracy of 94.80% and a testing accuracy of 98.00%. The contribution of this research is to accurately detect weather conditions by providing relevant information to the driver or an automated control system, thereby increasing cybersecurity and driving comfort.

Keywords: *autonomous car*, CNN, ResNet18, surveillance systems, weather condition detection

## 1. PENDAHULUAN

Sistem bantuan pengemudi yang canggih merupakan hal yang umum di dunia otomotif saat ini [1]. Kemampuan mengemudi dalam berbagai kondisi lingkungan merupakan masalah penting untuk sistem mengemudi otomatis pada *autonomous car* [2]. Kondisi cuaca biasanya dapat mengganggu atau menurunkan fungsi sistem transportasi dengan bantuan penglihatan saja [3]. Pemrosesan gambar dapat digunakan untuk menentukan kondisi cuaca [4]. Sistem keselamatan yang aktif pada otomotif canggih, secara umum, dan termasuk kendaraan *autonomous*, sangat bergantung pada data visual untuk mengklasifikasikan dan melokalisasi objek guna membantu kendaraan yang bersangkutan melakukan manuver dengan aman di lingkungannya [5].

Kondisi cuaca buruk dianggap sebagai pembatasan fungsi kamera yang berbahaya, yang berdampak serius pada kinerja algoritme visi

komputer yang diadopsi untuk pemahaman pemandangan dalam pemandangan lalu lintas [6]. Mengotomatiskan pemantauan cuaca tidak hanya memungkinkan, tetapi juga bermanfaat [7]. Pengenalan otomatis kondisi cuaca memiliki nilai aplikasi yang penting dalam peringatan kondisi lalu lintas, penggerak tambahan mobil, sistem transportasi cerdas, dan aspek lainnya [8]. Selain itu, serangan siber dapat terjadi dengan mengubah kondisi cuaca sensor atau gambar secara tiba-tiba, dan menjadi perhatian khusus penelitian ini dari sisi keamanan siber.

Sistem deteksi yang diimplementasi pada kendaraan dapat meningkatkan akurasi deteksi dan visibilitas dalam berbagai kondisi cuaca [9]. Salah satu cara yang dapat dilakukan dengan mengimplementasi sistem klasifikasi cuaca berdasarkan kamera pada *autonomous car* untuk membatasi kecepatannya. Penelitian ini mengajukan model ResNet18 yang dibandingkan dengan model

CNN yang telah banyak diaplikasikan pada sistem pendeteksi cuaca. Dengan mengimplementasikan sistem ini, diharapkan *autonomous car* memiliki sistem pengawasan untuk meningkatkan keamanan [10] dan kenyamanan dalam berkendara. Sehingga, model ResNet18 dapat menjadi kontribusi penelitian untuk mendeteksi kondisi cuaca secara akurat dengan memberikan informasi yang relevan kepada pengemudi atau sistem pengendalian otomatis, sehingga meningkatkan keamanan siber dan kenyamanan dalam berkendara.

## 2. LANDASAN TEORI

Beberapa peneliti telah melakukan penelitian untuk mengklasifikasikan gambar cuaca dan studi ini mengimplementasikannya ke dalam sistem keamanan *autonomous car*. Dengan implementasi ini diharapkan pengguna *autonomous car* mendapatkan keamanan dan kenyamanan dalam berkendara.

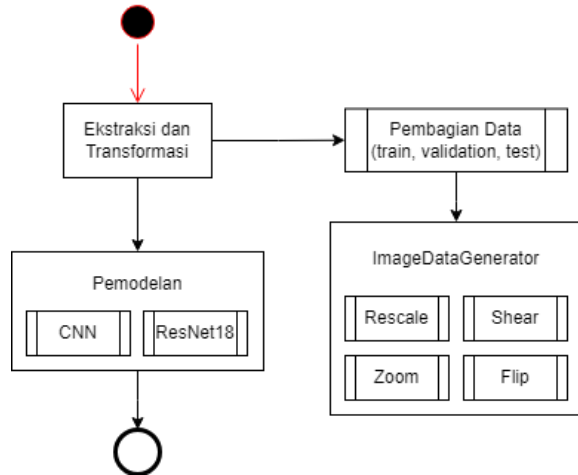
Elhoseiny *et al.* [11] pada penelitiannya melakukan klasifikasi pada gambar cuaca dengan tingkat akurasi 82,2% menggunakan model klasifikasi gambar CNN. Pengklasifikasian gambar lainnya yang dirancang dan dikembangkan oleh Sharma and Ismail [12] menggunakan CNN menunjukkan akurasi, akurasi validasi, *losses*, dan *validation losses* masing-masing sekitar 94%, 92%, 18%, dan 22%. Kondisi permukaan jalan di musim dingin juga diklasifikasi oleh Zhang *et al.* [13] menggunakan CNN, yang menghasilkan rata-rata *F1-Score model dual-stream* adalah 0.866, 0.935, 0.985, dan 0.888 pada kondisi permukaan bersalju, es, basah, dan berlumpur.

Sedangkan penelitian ResNet18 oleh Al-Haija *et al.* [14] diusulkan sebagai model *autonomous IoT* yang mengklasifikasikan gambar cuaca dengan akurasi mencapai 98.22%. Arsitektur ResNet18 menghasilkan kinerja terbaik dalam hal pendeteksian cuaca dan kondisi permukaan secara keseluruhan yang mengesankan, dengan akurasi sekitar 97% dan 99% [15].

Dari berbagai penelitian yang ada, belum adanya pengimplementasian sistem pembatasan kecepatan *autonomous car* dari klasifikasi gambar cuaca. Sistem pembatasan kecepatan menjadi sangat relevan dan penting dalam pengembangan *autonomous car*, terutama ketika menghadapi kondisi cuaca yang berkabut atau hujan. Dalam kondisi cuaca tersebut, visibilitas dapat berkurang secara signifikan, dan objek-objek di jalan, seperti kendaraan lain, pejalan kaki, atau hewan, dapat sulit terlihat dan diidentifikasi dengan jelas. Sehingga penelitian ini akan membandingkan performa dari model CNN dan ResNet18 dalam mengklasifikasikan gambar berdasarkan kondisi cuaca yang terdeteksi. Model terbaik kemudian akan diimplementasi pada sistem *autonomous car* sebagai sistem pengawasan untuk meningkatkan keamanan dan kenyamanan dalam berkendara dengan pembatasan kecepatan.

## 3. METODE PENELITIAN

Pada penelitian ini digunakan metode eksperimen. Tahapan pada metode ini terbagi menjadi 3 proses, yaitu ekstraksi dan transformasi data, pemodelan CNN, dan pemodelan ResNet18 seperti yang dapat dilihat pada Gambar 1.



Gambar 1. Metode Eksperimen

Setelah ditemukan model terbaik antara CNN dan ResNet18, proses selanjutnya adalah implementasi yang dilakukan pada sistem. Implementasi bertujuan untuk mengklasifikasikan cuaca hasil tangkapan gambar dari kamera yang terpasang pada mobil.

### 3.1. Ekstraksi dan Transformasi Data

Proses pertama yang dilakukan pada tahapan ini adalah pembagian data terlebih dahulu. Data yang berjumlah 1,200 gambar dibagi ke dalam 4 kelas, yaitu *cloudy* (mendung), *foggy* (berkabut), *rain* (hujan), dan *sunrise* (cerah). Kemudian, data dibagi menjadi 3 bagian yaitu: data latih (*train*), data validasi (*validation*), dan data uji (*test*) dengan proporsi 80:10:10. Setelah semua data terbagi, proses selanjutnya adalah melakukan ekstraksi dan transformasi data menggunakan fungsi 'ImageDataGenerator' dengan proses sebagai berikut.

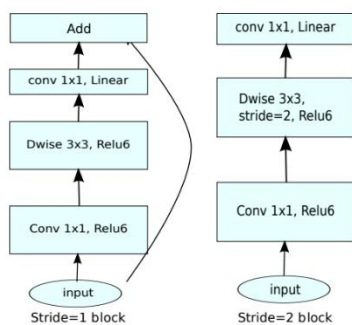
- Rescale*, fungsi 'rescale=1./255' digunakan untuk mereskalasi (*rescale*) intensitas piksel dalam gambar. Dalam hal ini, setiap nilai piksel dijadikan dalam rentang antara 0 hingga 1 dengan membagi setiap nilai piksel dengan 255 [16]. Ini adalah langkah umum dalam pra-pemrosesan gambar karena intensitas piksel biasanya berada dalam rentang 0 hingga 255.
- Shear*, fungsi 'shear\_range=0.2' digunakan untuk mengontrol transformasi *shear* (pergeseran) gambar. *Shear* merupakan operasi pergeseran satu sisi gambar tanpa mengubah ukuran gambar [17].
- Zoom*, fungsi 'zoom\_range=0.2' digunakan untuk mengontrol tingkat perbesaran atau perkecilan gambar (*zoom*) [18]. Dalam hal ini, gambar dapat diperbesar hingga 20% atau diperkecil hingga 20%.

- d. *Flip*, fungsi 'horizontal\_flip=True' digunakan untuk mengatur apakah akan dilakukan *flipping* horizontal pada gambar. *Flipping* horizontal adalah operasi membalikkan gambar secara horizontal seperti mencerminkan gambar dari sumbu vertikal [19].

Fungsi tersebut adalah penggunaan objek ImageDataGenerator dalam pemrosesan gambar pada sebuah tugas pelatihan (*training*), validasi (*validation*), dan pengujian (*testing*) pada sebuah model machine learning yang mungkin menggunakan DL. Penggunaan ImageDataGenerator ini sangat berguna dalam melatih model *deep learning*, karena dapat memperkaya *dataset* dan membuat model lebih tahan terhadap variasi dalam data gambar yang dihadapinya. Dengan melakukan augmentasi data seperti ini, model cenderung menjadi lebih umum dan dapat meningkatkan kemampuan generalisasinya.

### 3.2. Pemodelan CNN

Model CNN menggunakan base\_model dari MobileNetV2 dimana tiap *layer* di *training* berdasarkan weight dari *pre-trained imagenet*. Model CNN tersebut akan menggunakan MobileNetV2 sebagai dasar untuk ekstraksi fitur gambar. Selanjutnya, model ini akan disesuaikan dengan dataset khusus untuk tugas tertentu, dan proses training akan mengatur ulang bobot di setiap lapisan agar model lebih terfokus pada tugas klasifikasi atau deteksi objek yang diinginkan, dengan memanfaatkan pengetahuan yang sudah dipelajari dari *dataset* ImageNet sebelumnya. Pendekatan ini memungkinkan model untuk memiliki kemampuan generalisasi yang baik dan menghasilkan performa yang lebih baik dalam memecahkan masalah gambar yang lebih spesifik.



Gambar 2. Arsitektur Model CNN – MobileNetV2 [20]

Penelitian ini menggunakan arsitektur model seperti pada penelitian Sandler *et al.* [20], yaitu arsitektur MobileNetV2 yang dapat dilihat pada Gambar 2. MobileNetV2 berisi lapisan konvolusi penuh awal dengan 32 filter, diikuti oleh 19 lapisan bottleneck sisa. Digunakan ReLU6 sebagai non-linearitas karena ketangguhannya saat digunakan dengan komputasi presisi rendah. Ukuran kernel yang digunakan adalah  $3 \times 3$  sebagai standar untuk jaringan modern, dan menggunakan dropout dan normalisasi

batch selama pelatihan.

Dengan pengecualian pada lapisan pertama, digunakan tingkat ekspansi yang konstan di seluruh jaringan. Dalam percobaan kami, kami menemukan bahwa tingkat ekspansi antara 5 dan 10 menghasilkan kurva kinerja yang hampir sama, dengan yang lebih kecil jaringan yang lebih kecil lebih baik dengan tingkat ekspansi yang sedikit lebih kecil dan jaringan yang lebih besar memiliki kinerja yang sedikit lebih baik dengan tingkat ekspansi yang lebih besar.

Pada semua percobaan, digunakan faktor ekspansi sebesar 6 yang diterapkan pada ukuran tensor input. Sebagai contoh untuk lapisan *bottleneck* yang mengambil tensor input 64 saluran dan menghasilkan tensor dengan 128 saluran, maka lapisan ekspansi menengah adalah  $64 - 6 = 384$  saluran.

### 3.3. Pemodelan ResNet18

ResNet18 memiliki 18 lapisan (termasuk lapisan konvolusi dan lapisan *fully connected*) dan telah terbukti efektif dalam banyak tugas pengolahan gambar. *Pre-trained* model yang sudah optimal dari PyTorch merujuk pada model ResNet18 yang telah dilatih sebelumnya pada dataset yang besar dan beragam menggunakan PyTorch. Model ini telah melewati fase pelatihan (*training*) yang memakan waktu dan memanfaatkan banyak data gambar dari dataset seperti ImageNet untuk menyesuaikan bobot dan parameter di setiap lapisan agar mampu mengenali berbagai fitur dan objek dalam gambar dengan baik. Dengan menambahkan fungsi aktivasi LogSoftmax pada model, hasil *output* dari model ResNet18 akan berupa probabilitas logaritmik dari tiap kelas, yang kemudian dapat digunakan untuk melakukan klasifikasi multikelas. Dengan demikian, model ini dapat menghasilkan prediksi dengan lebih efisien dan akurat untuk berbagai kategori gambar yang telah dilatih sebelumnya pada dataset yang besar menggunakan PyTorch.

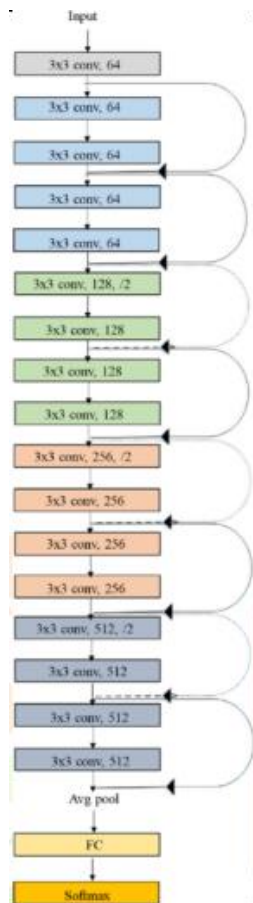
layer name	output size	18-layer
conv1	$112 \times 112$	
conv2_x	$56 \times 56$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
	$1 \times 1$	
FLOPs		$1.8 \times 10^9$

Gambar 3. Arsitektur Model ResNet18 - ImageNet [21]

Model yang diajukan menggunakan arsitektur model yang digunakan Abhishek *et al.* [21], yang dapat dilihat pada Gambar 3 dengan detail arsitektur dapat dilihat pada Gambar 4. ResNet18 memiliki 18 lapisan dengan kernel  $7 \times 7$  sebagai lapisan pertama.

Model ini memiliki empat lapisan ConvNets yang identik. Setiap lapisan terdiri dari dua blok sisa. Setiap blok terdiri dari dua lapisan bobot dengan koneksi lompatan yang terhubung ke *output* dari lapisan bobot kedua dengan ReLU. Jika hasilnya sama dengan input dari lapisan ConvNet, maka koneksi identitas digunakan. Tetapi, jika input tidak sama dengan *output*, maka *convolutional pooling* dilakukan pada koneksi selanjutnya. Ukuran input yang diambil oleh ResNet18 adalah (224, 224, 3), yang dilakukan dengan menerapkan augmentasi menggunakan *library* AugStatic pada pra-pemrosesan. Pada lapisan (224, 224, 3) dengan 224 adalah lebar dan tinggi, 3 adalah saluran *red-blue-green* (RGB). *Output* yang dihasilkan adalah *Fully Connected layer* yang memberikan input kepada *sequential layer*.

Dalam metodologi yang diusulkan, ada lapisan sekuensial tambahan pada akhir lapisan terakhir dari ResNet18. Masukan akan dilewatkan ke linear (512,512), yang keluarannya diumpungkan ke fungsi aktivasi ReLU pertama. Kemudian lapisan *dropout* (0,2) digunakan, diikuti oleh lapisan linear (512, 2). Terakhir, dilewatkan melalui LogSoftmax untuk mendapatkan logaritma dari probabilitas. Model ini menggunakan kerangka kerja Pytorch.



Gambar 4. Detail Arsitektur Model ResNet18 – ImageNet [21]

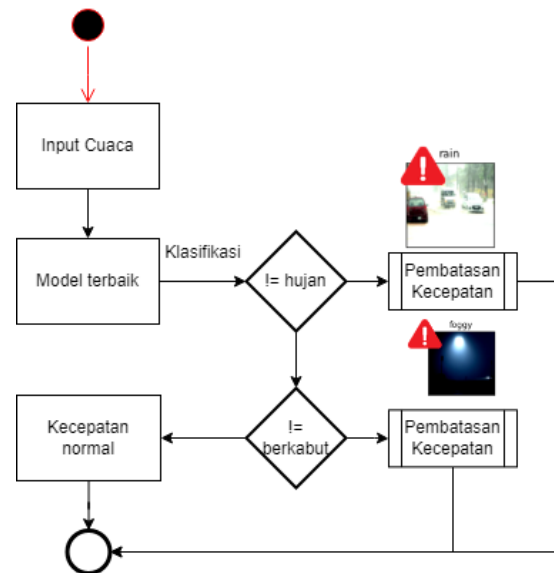
### 3.4. Implementasi

Setelah proses pemodelan selesai dan ditemukan

model terbaik antara CNN dengan ResNet18, maka proses selanjutnya adalah implementasi. Model terbaik pada Gambar 5 diterapkan pada miniPC yang terhubung pada kamera untuk menangkap gambar cuaca secara langsung. Gambar cuaca yang menjadi input kemudian diklasifikasikan oleh model.

Apabila tidak ditemukan klasifikasi hujan atau berkabut, maka kecepatan mobil diatur dalam posisi normal oleh sistem. Namun, jika terdeteksi klasifikasi gambar tersebut adalah hujan atau berkabut, maka sistem akan mengatur batasan kecepatan sesuai aturan yang berlaku di wilayah tersebut.

Dalam hal peningkatan keamanan siber pada sistem, proses perubahan pembatasan kecepatan pada sistem pengawasan *autonomous car* dilakukan dengan penambahan *delay* yang dapat disesuaikan. Sehingga sistem tersebut tidak membuat celaka pengunanya.



Gambar 5. Proses Implementasi Model Terbaik

## 4. HASIL DAN PEMBAHASAN

Pada tahap training, dilakukan penentuan parameter penting untuk setiap model CNN dan ResNet18 guna memastikan pelatihan yang efektif dan efisien. Parameter-parameter tersebut adalah jumlah epoch, ukuran gambar (dimensi gambar), dan *batch\_size*.

Pertama, untuk setiap model CNN dan ResNet18, pelatihan dilakukan dengan menggunakan 50 epoch. Ini berarti proses pelatihan akan berlangsung sebanyak 50 iterasi di atas seluruh dataset pelatihan. Lebih banyak epoch biasanya memberikan lebih banyak kesempatan bagi model untuk belajar pola yang lebih kompleks dan dapat meningkatkan performa.

Kedua, *batch\_size* pada model CNN dan ResNet18 diatur sebesar 16. *Batch\_size* menentukan berapa banyak sampel yang akan diproses dalam setiap iterasi pelatihan. Dalam kasus ini, 16 gambar akan digunakan bersamaan dalam satu iterasi.

Penggunaan `batch_size` yang sesuai dapat mengoptimalkan penggunaan sumber daya komputasi dan mempercepat waktu pelatihan.

Ketiga, ukuran gambar (dimensi gambar) yang digunakan adalah 150x150 piksel. Ini menunjukkan bahwa gambar dalam dataset akan diubah atau dipangkas menjadi ukuran 150x150 piksel sebelum dimasukkan ke dalam model. Memilih ukuran gambar yang sesuai adalah penting untuk konsistensi data dan juga mempengaruhi kompleksitas dan kinerja model.

Tabel 1. Evaluasi Performa Model

Model	Pelatihan	Pengujian
CNN	Akurasi = 84.80% Loss = 61.56%	Akurasi = 84.00% Loss = 53.63%
ResNet18	Akurasi = 94.80% Loss = 4.66%	Akurasi = 98.00% Loss = 0.7%

Pada Tabel 1 dapat diketahui bahwa model CNN memiliki hasil evaluasi akurasi yang cukup baik, yaitu sekitar 84.80% pada tahap pelatihan dan 84.00% pada tahap pengujian. Namun, *loss* pada tahap pelatihan dan pengujian masih tinggi, yaitu 61.56% dan 53.63% masing-masing. *Loss* yang tinggi menunjukkan bahwa model ini masih memiliki masalah dalam menyesuaikan diri dengan data pelatihan dan data pengujian. Hal ini dapat mengindikasikan adanya *overfitting*, yaitu model terlalu khusus pada data pelatihan dan tidak dapat umum pada data baru.

Hasil studi yang dilakukan pada klasifikasi cuaca mengenai model ResNet18 memiliki keunggulan yang sama baiknya dengan penelitian Al-Haija *et al.* [14] dan Khan and Ahmed [15]. Model ResNet18 menunjukkan performa yang sangat baik. Pada tahap pelatihan, akurasi mencapai 94.80%, yang menunjukkan bahwa model ini berhasil dengan sangat baik dalam mempelajari data pelatihan. Selain itu, *loss* pada tahap pelatihan sangat rendah, yaitu hanya sebesar 4.66%. Hal ini menandakan bahwa model ini telah dapat menemukan pola dan fitur yang baik pada data pelatihan.

```
Test Loss: 0.007007
Test Accuracy of cloudy: 100% (51/51)
Test Accuracy of foggy: 100% (69/69)
Test Accuracy of rain: 98% (56/57)
Test Accuracy of sunrise: 95% (60/63)
Test Accuracy (Overall): 98% (236/240)
```

Gambar 6. Hasil Evaluasi Pengujian ResNet18

Pada tahap pengujian, Gambar 6 menunjukkan performa model ResNet18 semakin meningkat dengan akurasi mencapai 98.00%, yang mengindikasikan bahwa model ini memiliki kemampuan generalisasi yang sangat baik dan mampu dengan baik memprediksi data yang belum pernah dilihat sebelumnya. Selain itu, *loss* pada tahap pengujian sangat rendah, yaitu hanya 0.7%, menunjukkan bahwa model ini dapat menghasilkan prediksi yang sangat akurat pada data pengujian.

Pada Gambar 7 dapat dilihat model memprediksi dan mengklasifikasikan gambar dengan baik dan akurat. Ketika ditemukan hasil klasifikasi hujan (*rain*)

dan berkabut (*foggy*) secara otomatis sistem akan memberikan peringatan dan membatasi kecepatan *autonomous car*. Pembatasan kecepatan dapat disesuaikan dengan kondisi dan kebutuhan masing-masing negara yang mengaturnya. Hal ini dapat memberikan keamanan dan kenyamanan ke pengguna *autonomous car*. Dalam hal peningkatan keamanan siber pada sistem, proses perubahan pembatasan kecepatan pada sistem pengawasan *autonomous car* dilakukan dengan penambahan *delay* yang dapat disesuaikan. Sehingga sistem tersebut tidak membuat celaka penggunaannya.



Gambar 7. Hasil Evaluasi Validasi ResNet18

Pembatasan kecepatan sudah diatur dalam Peraturan Menteri Perhubungan Nomor 111 Tahun 2015 yang dijelaskan Setiawan *et al.* [22] dan Plue *et al.* [23], yaitu pada jalan bebas hambatan, ditentukan batas kecepatan paling rendah 60 kilometer per jam dalam kondisi arus bebas dan paling tinggi 100 kilometer per jam. Pada jalan antarkota, batas kecepatan yang ditentukan paling tinggi 80 kilometer per jam. Kawasan perkotaan batas kecepatan paling tinggi yaitu 50 kilometer per jam. Sedangkan, kawasan pemukiman paling tinggi 30 kilometer per jam.

Tabel 2. Kecepatan Rata-rata Kendaraan Indonesia di Tol (km/jam) [24]

Kendaraan	Minimum	Maksimum	Rata-rata
Mobil	57	94	88
Bus	52	86	72
Truk	32	70	62

Acuan standar batasan kecepatan kendaraan Indonesia pada jalan Tol telah diteliti R. I. Mauliza *et al.* [24] yang dapat dilihat pada Tabel 2. Jika ingin diimplementasikan ke dalam sistem pembatasan kecepatan *autonomous car* di Indonesia, penelitian tersebut dapat menjadi landasan awal. Meskipun pada penelitian selanjutnya diharapkan acuan kecepatan kendaraan tidak hanya dilakukan di jalan Tol. Jarak pengereman juga perlu diperhatikan seperti yang tercantum dalam penelitian Noesaku da Costa *et al.* [25], bahwa jarak rata-rata minimum kendaraan sebesar 4 meter dan jarak rata-rata maksimum pengereman kendaraan sebesar 6 meter.

## 5. KESIMPULAN

Model ResNet18 menunjukkan performa yang jauh lebih baik daripada model CNN pada *dataset* yang digunakan. Model ResNet18 memiliki akurasi yang tinggi dan *loss* yang rendah pada tahap pelatihan dan pengujian, menunjukkan kemampuan model untuk belajar dan menggeneralisasi dengan baik.



Sebaliknya, model CNN memiliki akurasi yang lebih rendah dan *loss* yang lebih tinggi, menandakan adanya *overfitting* dan kemampuan generalisasi yang kurang baik. Oleh karena itu, dalam kasus ini, model ResNet18 adalah pilihan yang lebih baik untuk memecahkan masalah yang diberikan dalam pengawasan sistem pembatasan kecepatan *autonomous car* untuk meningkatkan keamanan siber pada sistemnya dan kenyamanan penggunaannya.

Pada masa mendatang, diperlukan suatu sistem untuk membatasi kecepatan yang tidak hanya pada ruang lingkup jalan Tol. Namun, dapat diterapkan pada berbagai kondisi jalan yang ada dan tetap pada aturan berkendara di wilayah Indonesia.

## REFERENSI

- [1] A. Cord and D. Aubert, "Towards rain detection through use of in-vehicle multipurpose cameras," 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 2011, pp. 833-838, doi: 10.1109/IVS.2011.5940484.
- [2] K. Yoneda, N. Suganuma, R. Yanase and M. Aldibaja, "Automated driving recognition technologies for adverse weather conditions," *IATSS Research*, 2019, pp. 253-262. <https://doi.org/10.1016/J.IATSSR.2019.11.005>.
- [3] L. -W. Kang, K. -L. Chou and R. -H. Fu, "Deep Learning-Based Weather Image Recognition," 2018 International Symposium on Computer, Consumer and Control (IS3C), Taichung, Taiwan, 2018, pp. 384-387, doi: 10.1109/IS3C.2018.00103.
- [4] D. A. Padilla, J. F. Villaverde, J. J. T. Magdaraog, A. J. L. Oconer and J. P. Ranjo, "Vehicle and Weather Detection Using Real Time Image Processing Using Optical Flow and Color Histogram," 2019 5th International Conference on Control, Automation and Robotics (ICCAR), Beijing, China, 2019, pp. 880-883, doi: 10.1109/ICCAR.2019.8813346.
- [5] M. Hnawa and H. Radha, "Object Detection Under Rainy Conditions for Autonomous Vehicles: A Review of State-of-the-Art and Emerging Techniques," in *IEEE Signal Processing Magazine*, vol. 38, no. 1, pp. 53-67, Jan. 2021, doi: 10.1109/MSP.2020.2984801.
- [6] KENK, Mourad (2020), "DAWN", Mendeley Data, V3, doi: 10.17632/766ygrbt8y.3
- [7] V. Abhyankar, A. G. Singh, P. Paul, A. Mehta and S. Vidhya, "Portable Autonomous Rain Prediction Model Using Machine Learning Algorithm," 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, 2019, pp. 1-4, doi: 10.1109/ViTECoN.2019.8899704.
- [8] Jayanthi, K. B., K. Vanitha, M. Malathi, J. Chandrasekar, and R. Khavva, "An Efficient Weather Recognition Algorithm on Highway Roads for Vehicle Guidance," *International Journal of Health Sciences*, vol. 6, no. S8, 2022, pp. 1212-23, doi:10.53730/ijhs.v6nS8.9882.
- [9] X-Z. Chen, C-M. Chang, C-W. Yu and Y-L. Chen, "A Real-Time Vehicle Detection System under Various Bad Weather Conditions Based on a Deep Learning Model without Retraining," *Sensors* 20, 2020, no. 20: 5731. <https://doi.org/10.3390/s20205731>.
- [10] Z. Paschalidi, W. Acevedo, M. Hellweg, T. Kratzsch, R. Potthast and J. Nachtigall, "Real time weather for autonomous driving and precise road weather forecasts based on floating car data and seamless integration of Ultra-Rapid Data Assimilation and Nowcasting," EMS Annual Meeting 2021, online, 2021, EMS2021-385, <https://doi.org/10.5194/ems2021-385>.
- [11] M. Elhoseiny, S. Huang and A. Elgammal, "Weather classification with deep convolutional neural networks," 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 2015, pp. 3349-3353, doi: 10.1109/ICIP.2015.7351424.
- [12] A. Sharma and Z. S. Ismail, "Weather Classification Model Performance: Using CNN, Keras-Tensor Flow," ITM Web Conf., 2022, doi: 10.1051/itmconf/20224201006.
- [13] C. Zhang, E. Nateghinia, L. F. Miranda-Moreno and L. Sun, "Winter road surface condition classification using convolutional neural network (CNN): visible light and thermal image fusion," *Canadian Journal of Civil Engineering*, 2021, pp. 569-578, doi: 10.1139/cjce-2020-0613.
- [14] Q. A. Al-Haija, M. A. Smadi and S. Zein-Sabatto, "Multi-Class Weather Classification Using ResNet-18 CNN for Autonomous IoT and CPS Applications," 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2020, pp. 1586-1591, doi: 10.1109/CSCI51800.2020.00293.
- [15] M. N. Khan and M. M. Ahmed, "Weather and surface condition detection based on road-side webcams: Application of pre-trained Convolutional Neural Network," *International Journal of Transportation Science and Technology*, 2022, pp. 468-483, doi: 10.1016/j.ijtst.2021.06.003.
- [16] Z. Iklima, T. M. Kadarina, and R. Priambodo, "Reduction of Feature Extraction for COVID-19 CXR using Depthwise Separable Convolution Network", *j.electron.electromedical.eng.med.inform*, vol. 4, no. 4, pp. 204-209, Oct. 2022, doi: 10.35882/jeeemi.v4i4.255.
- [17] F. Rajeena P. P. et al., "A Novel Method for the Classification of Butterfly Species Using Pre-Trained CNN Models," *Electronics*, vol. 11, no.

- 13, p. 2016, Jun. 2022, doi: 10.3390/electronics11132016.
- [18] H. P. Kekal and D. U. E. Saputri, "Optimization of Melanoma Skin Cancer Detection with the Convolutional Neural Network", *MEDINFTEch*, vol. 1, no. 2, pp. 53–58, Jun. 2023, doi: 10.37034/medinftech.v1i2.10.
- [19] MD. T. Mallick. et al., "Deep learning based automated disease detection and pest classification in Indian mung bean," *Multimed Tools Appl* 82, pp. 12017–12041, 2023, <https://doi.org/10.1007/s11042-022-13673-7>.
- [20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *arXiv:1801.04381v4*, 2019.
- [21] A. V. S. Abhishek, V. R. Guralla and L. Sahoo, "ResNet18 Model With Sequential Layer For Computing Accuracy On Image Classification Dataset," *International Journal of Creative Research Thoughts (IJCRT)*, 2022.
- [22] D. Setiawan. Et al., "Implementasi teknik pwm pada rancang bangun alat deteksi kecepatan kendaraan berdasarkan perputaran roda berbasis mikrokontroller," *Jurnal SAINTIKOM*, 2020, doi: 10.53513/jis.v19i1.224.
- [23] V. H. A. Plue, D. G. N. da Costa and A. H. Pattiraja, "Analisis Batas Kecepatan Pada Jalan Lokal Sekunder," *ETERNITAS : Jurnal Teknik Sipil*, 2022, doi: 10.30822/eternitas.v2i1.1704.
- [24] R. I. Mauliza, T. B. Sabrina and W. Maulana, "Pelanggaran Kecepatan Kendaraan pada Ruas Jalan Tol Cipularang," *RekaRacana: Jurnal Teknik Sipil*, 2019, doi: 10.26760/rekaracana.v5i1.39.
- [25] D. G. Noesaku da Costa, O. Edvict Semiun, dan A. . Liufeto, "Manajemen Kecepatan dan Jarak Aman Antar Kendaraan Berdasarkan Perbedaan Kecepatan Kendaraan yang Berjalan Beriringan", *JTS ITP*, vol. 7, no. 2, hlm. 3, Jul 2020, doi: 10.21063/jts.2020.V702.03.