

IMPLEMENTASI METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) UNTUK KLASIFIKASI GAMBAR MOBIL DAN MOTOR MENGUNAKAN KERAS

Abdillah Nur Ridho¹, Ghalda Mellyka², Fadil Saputra³, Amin Padmo Azam Masa⁴

Program Studi Sistem Informasi, Fakultas Teknik, Universitas Mulawarman

ABSTRAK

Penggunaan kendaraan yang semakin masif membuat beberapa pengendara motor sering kali menerobos jalur yang tidak semestinya, contohnya di Jembatan Mahkota IV (kembar) Samarinda di mana kendaraan motor menerobos jalur mobil dikarenakan alasan ingin cepat sampai dan medan jalur motor yang sempit dan berlubang. Oleh karena itu, dilakukan implementasi dan klasifikasi citra menggunakan metode *convolutional neural network* (CNN) menggunakan keras dengan hasil dilihat dari nilai *loss* pada iterasi ke 20 pada data latih yaitu 0.119171 dan nilai *accuracy* pada data latih yaitu 0.960106. Hasil pengujian pada data uji menggunakan *library* keras yang terdiri dari 12 citra terbagi menjadi 6 mobil dan juga 6 motor dapat dilihat pada Gambar 6. Berdasarkan pengujian model yang dilakukan terhadap data uji, 6 motor berhasil diklasifikasi dengan benar dan 6 mobil berhasil diklasifikasi dengan benar. Dengan demikian model mampu memprediksi dengan baik yaitu akurasi 100% secara keseluruhan dari data uji yang sudah disiapkan.

Kata Kunci: *Convolutional Neural Network, Keras, Kendaraan.*

PENDAHULUAN

Kendaraan merupakan salah satu sarana angkut di jalan yang terdiri atas kendaraan bermotor dengan kendaraan tidak bermotor menurut UU Nomor 22 Tahun 2009 Pasal 1 Angka 7. Salah satu jenis kendaraan yang sering digunakan ialah kendaraan roda dua yang disebut dengan motor, lalu kendaraan roda empat yang disebut mobil. Kedua jenis kendaraan ini sering sekali digunakan oleh masyarakat Indonesia, terkhususnya di kota-kota besar seperti kota-kota besar seperti Kota Samarinda. Penggunaan kendaraan yang semakin masif sehingga membuat beberapa pengendara motor sering kali menerobos jalur yang tidak semestinya, contohnya di Jembatan Mahkota IV (Kembar) Samarinda yang di mana sudah disiapkan 2 jalur agar tidak terjadi kecelakaan pengguna kendaraan mobil dengan motor. Namun dengan adanya 2 jalur ini

malah membuat beberapa persoalan, seperti halnya kendaraan motor menerobos jalur mobil dikarenakan alasan ingin cepat sampai dan medan jalur motor yang sempit dan berlubang.

Pengenalan jenis kendaraan dapat dilakukan dengan metode *deep learning*, beberapa penelitian mengenai pengolahan citra dengan menggunakan metode *convolutional neural network* mendapatkan hasil akurasi, yaitu penelitian yang dilakukan oleh Fonda (2020) pada batik Riau dengan bukan batik Riau menghasilkan akurasi 65% disebabkan antara batik Riau dengan bukan batik Riau memiliki kesamaan motif tetapi perbedaan pada warna cerap. Selanjutnya penelitian metode *convolutional neural network* pada ekspresi manusia oleh Nugroho et al. (2020) dengan *epoch* 100, *batch size* 128 didapatkan hasil akurasi *training* sebesar 90% dan *validation* sebesar 65%. Lalu

hasil percobaan dari total 35 ekspresi, 28 ekspresi berhasil ditebak dengan benar dengan mendapatkan akurasi sebesar 80%. Penerapan algoritma *convolutional neural network* untuk klasifikasi kendaraan prioritas menunjukkan bahwa skenario terbaik terdapat pada skenario 2 dengan data training sebesar 60%, data *validation* sebesar 20%, dan data *testing* sebesar 20% berhasil mendapatkan *validation accuracy* sebesar 66,15% dan *testing accuracy* sebesar 69,231% yang penelitian ini dilakukan oleh Abdulhakim & Dermawan (2021).

Dari beberapa studi terdahulu, didapatkan hasil metode yang ingin digunakan peneliti adalah metode beberapa *convolutional neural network*. Selanjutnya dalam pengelolaan citra digital terdapat tahapan pemrosesan yaitu mulai dari akuisisi citra, pengolahan citra dengan perbaikan kualitas citra dan segmentasi citra, pengenalan pola dengan ekstraksi ciri dan klasifikasi citra. Pengolahan kali ini dilakukan untuk menentukan klasifikasi kendaraan antara mobil dengan motor yang mengambil studi kasus di Jembatan Mahkota IV (Kembar), Samarinda karena banyaknya pengemudi motor yang menerobos jalur mobil pada jembatan.

TINJAUAN PUSTAKA

Convolutional Neural Network

Convolutional Neural Network (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang termasuk dalam *neural network* bertipe *feed forward* (tidak berulang). *Convolutional Neural* didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *deep neural network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra (Putra, 2016).

Convolutional Neural Network (CNN) merupakan jenis arsitektur jaringan saraf yang khusus dirancang

untuk mengolah citra dan menemukan pola di dalamnya. CNN dilakukan dengan mengakuisisi citra sebagai *input* melalui beberapa lapisan khusus. Lapisan yang ada di dalam CNN meliputi konvolusi, aktivasi, *pooling*, dan *fully connected layer*. Selama proses pelatihan model, CNN belajar dari dataset citra dengan mencocokkan fitur-fitur pada citra dengan label yang sesuai. Dengan menggunakan algoritma optimisasi, CNN memperbarui parameter-parameter internalnya untuk meningkatkan kemampuan.

Deep Learning

Deep Learning atau *deep structured learning* atau *hirarchical learning* atau *deep neural* merupakan metode *learning* yang memanfaatkan *multiple non-linier transformation*, *deep learning* dapat dipandang sebagai gabungan *machine learning* dengan AI (*artificial neural network*) (Primartha, 2018)

Deep Learning merupakan salah satu pembelajaran mesin yang berhubungan dengan arsitektur jaringan saraf tiruan (*neural networks*) yang mirip seperti kerja otak manusia di mana neuron terintegrasi satu sama lain dan membentuk jaringan neuron yang rumit. Beberapa algoritma yang masuk ke dalam *deep learning* ialah *Convolutional Neural Networks* (CNN), *Recurrent Neural Networks* (RNN), *Long Short-Term Memory* (LSTM), dan sebagainya.

Keras

Seiring dengan banyaknya pengembangan dan riset tentang *Deep Learning*, banyak *library* yang bermunculan dengan fokus mempelajari tentang jaringan syaraf tiruan. salah satu contohnya yaitu keras (Santoso & Gunawan Ariyanto, 2018). Keras adalah *library open-source* yang digunakan untuk membangun dan melatih jaringan saraf tiruan (*neural networks*) sehingga

mudah dalam mengimplementasikan ke dalam kode yang lebih sederhana dengan beberapa lapisan seperti lapisan konvolusi, lapisan rekurensi, dan lapisan penggabungan. Keras juga mendukung pelatihan paralel dan memungkinkan integrasi dengan TensorFlow untuk performa yang lebih tinggi.

Segmentasi Citra

Segmentasi citra merupakan tahapan dalam pengolahan citra digital yang memisahkan atau membagi suatu citra menjadi beberapa bagian yang berbeda berdasarkan fitur atau karakteristik tertentu. Segmentasi citra adalah proses pengolahan citra yang bertujuan memisahkan wilayah (*region*) objek dengan wilayah latar belakang agar objek mudah dianalisis dalam rangka mengenali objek yang banyak melibatkan persepsi visual (Destyningtias B. et al., 2010)

Klasifikasi Citra

Proses penemuan model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui (Annur, 2018). Selain itu, proses pengategorian objek pada suatu citra berdasarkan fitur-

fitur yang terdapat pada citra tersebut dan bertujuan untuk membedakan objek yang berbeda pada citra dengan tahapan pelatihan model dan evaluasi model yang selanjutnya diklasifikasikan untuk mengeluarkan prediksi terkait kategori yang sesuai dengan citra tersebut.

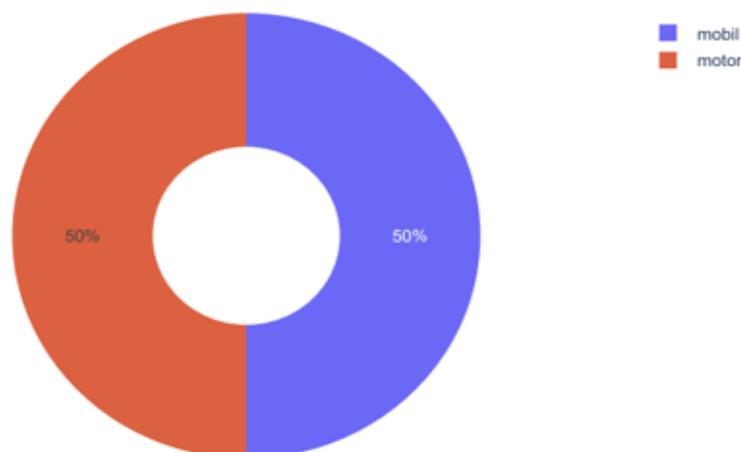
K-Means

Algoritma K-Means merupakan algoritma non hirarki yang berasal dari metode data *clustering*. Algoritma K-Means dimulai dengan pembentukan partisi kluster di awal kemudian secara interaktif partisi cluster ini diperbaiki hingga tidak terjadi perubahan yang signifikan pada partisi cluster (Sulistiyawati & Supriyanto, 2021).

METODE PENELITIAN

Pengumpulan Data Gambar

Pengumpulan data gambar dilakukan secara *online* atau bisa juga disebut dengan data daring. Data daring didapatkan dengan mengambil dari beberapa sumber yang ada di internet, dan data gambar yang di ambil adalah gambar motor dan mobil dengan berbeda-beda karakteristik dari segi jenis, warna, dan resolusi gambarnya. Jumlah data gambar yang di ambil adalah sebanyak 470 gambar dengan perbandingan 50:50.



Gambar 1. Distribusi Kelas

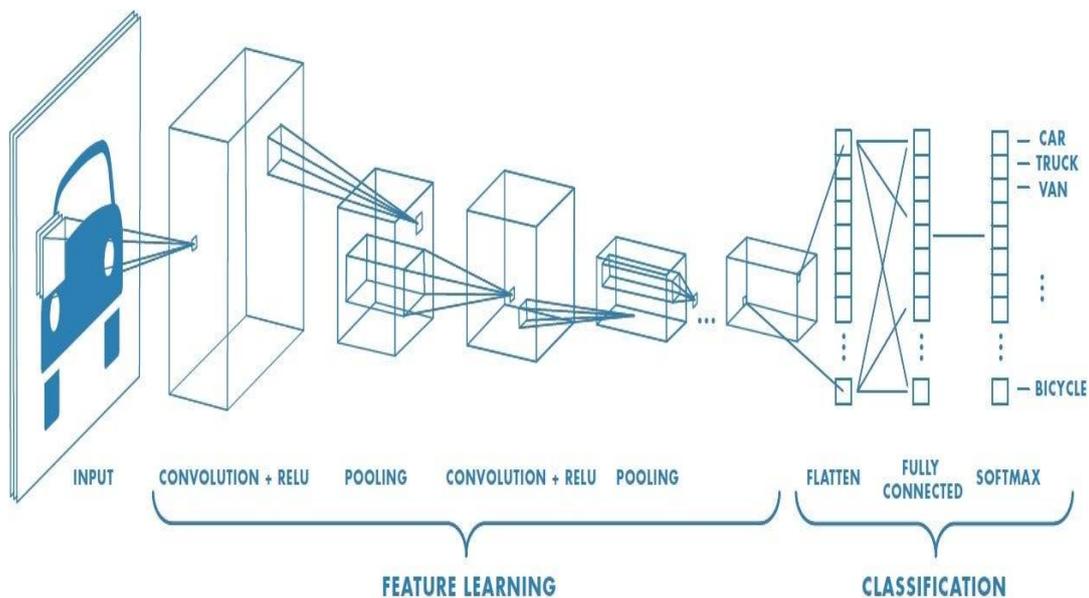
Pra-Pemrosesan Data

Pra-pemrosesan data adalah serangkaian langkah atau proses yang dilakukan pada data sebelum data tersebut digunakan untuk analisis, pemodelan, atau pemrosesan lebih lanjut. Pada tahapan ini dataset yang sudah disiapkan sebelumnya akan dilakukan pra-pemrosesan data gambar yaitu dengan cara melakukan *resizing* dan normalisasi, agar *output* yang dihasilkan memiliki ukuran gambar dan piksel yang sama untuk memastikan rentang piksel berada pada nilai 0 hingga 1. Kemudian data gambar yang sudah

tersedia tadi dilakukan *splitting* yaitu pemisahan antara data latih yang digunakan untuk melatih model dan data validasi digunakan untuk menyetel parameter dan evaluasi model.

Pembuatan Model CNN

Arsitektur model CNN adalah struktur atau tata letak lapisan-lapisan *neural network* yang dirancang khusus untuk memproses data gambar secara efektif. Arsitektur ini terdiri dari beberapa jenis lapisan yang bekerja bersama-sama untuk mengekstraksi fitur-fitur visual yang relevan dari gambar dan menghasilkan prediksi klasifikasi.



Gambar 2. Lapisan konvolusi

(Sumber: <https://www.trivusi.web.id/2022/04/algorithm-cnn.html>)

Berikut adalah yang beberapa jenis lapisan digunakan:

1. Lapisan Konvolusi (*Convolutional Layer*)

- Lapisan ini menggunakan operasi konvolusi untuk mengekstraksi fitur-fitur visual dari gambar.
- Setiap lapisan konvolusi terdiri dari beberapa *filter* yang menerapkan konvolusi pada gambar gambar untuk menghasilkan peta fitur.

2. Lapisan *Pooling* (*Pooling Layer*):

- Lapisan ini digunakan untuk mengurangi dimensi spasial dari peta fitur yang dihasilkan oleh lapisan konvolusi.
- *Pooling* dilakukan dengan operasi seperti *max pooling* atau *average pooling* untuk mengambil nilai maksimum atau rata-rata dalam jendela tertentu.

3. Lapisan Aktivasi (*Activation Layer*):

- Setelah lapisan konvolusi dan *pooling*, lapisan aktivasi diterapkan untuk memperkenalkan non-linearitas ke dalam model.
 - Aktivasi yang umum digunakan adalah ReLU (Rectified Linear Unit) untuk memperkuat fitur yang relevan.
4. Lapisan *Fully Connected (Fully Connected Layer)*:
- Setelah ekstraksi fitur melalui lapisan konvolusi dan *pooling*, fitur-fitur tersebut dikonversi menjadi vektor dan diberikan ke lapisan *fully connected*.
 - Lapisan *fully connected* bertindak sebagai klasifikasi akhir, menghubungkan fitur-fitur yang dihasilkan dengan kelas klasifikasi (mobil atau motor).
5. Lapisan *Output (Output Layer)*:
- Lapisan *output* menggunakan fungsi aktivasi seperti *softmax* untuk menghasilkan probabilitas kelas dari setiap contoh gambar.
 - Model akan mengklasifikasikan gambar sebagai mobil atau motor berdasarkan probabilitas tertinggi dari lapisan *output*.

Pelatihan Model

Setelah membangun arsitektur model, lalu dilakukan kompilasi model dengan menentukan *optimizer*, fungsi *loss*, dan metrik evaluasi. Pada penelitian kali ini, menggunakan *optimizer* 'adam' yang merupakan *optimizer* populer untuk pelatihan model *deep learning*. Kemudian fungsi *loss* yang digunakan adalah '*binary_crossentropy*' karena peneliti akan melakukan klasifikasi biner antara mobil dan motor. Dan metrik evaluasi yang digunakan adalah akurasi (*accuracy*) untuk memantau performa model selama pelatihan.

Setelah mengompilasi model, selanjutnya adalah melatih model menggunakan metode *fit*. Model dilatih dengan menggunakan data latih (*training set*) yang telah di-*generate* menggunakan *Image Data Generator*. Jumlah *epoch* (iterasi) pelatihan dapat disesuaikan sesuai kebutuhan, kali ini menggunakan 20 *epoch* agar nantinya bisa mendapatkan tingkat akurasi yang tinggi.

Selama proses pelatihan, model akan belajar dari data gambar dan mencoba untuk menemukan pola dan fitur-fitur yang membedakan antara mobil dan motor. Dengan melakukan iterasi melalui *epoch* dan melalui diharapkan dapat menghasilkan hasil yang baik dalam melakukan klasifikasi gambar mobil dan motor.

Evaluasi Model

Evaluasi model untuk implementasi metode *Convolutional Neural Network (CNN)* dalam klasifikasi gambar mobil dan motor menggunakan Keras dapat dilakukan dengan beberapa metrik evaluasi misalnya menggunakan akurasi, presisi, *recall*, dan *F1-score*. Akurasi mengukur sejauh mana model berhasil melakukan klasifikasi dengan benar secara keseluruhan, presisi mengukur sejauh mana prediksi positif yang dilakukan oleh model adalah benar, *recall* mengukur sejauh mana model dapat menemukan semua data yang sebenarnya positif, *F1-score* adalah metrik yang menggabungkan presisi dan *recall* untuk memberikan ukuran yang seimbang antara keduanya.

Pada evaluasi model kali ini peneliti menggunakan metrik evaluasi akurasi, yang mana metrik evaluasi akurasi ini dapat mengukur sejauh mana model berhasil melakukan klasifikasi yang benar secara keseluruhan. Akurasi dihitung dengan membagi jumlah prediksi yang benar dengan jumlah total data yang dievaluasi. Metrik ini

memberikan gambaran umum tentang performa keseluruhan model dalam mengenali gambar mobil dan motor.

HASIL DAN PEMBAHASAN

Pra-Pemrosesan Data

Setelah data terkumpul dan semua *package* yang dibutuhkan sudah diinstal, dilakukan data augmentasi untuk memperluas dataset dan meningkatkan keberagaman gambar yang digunakan dalam pelatihan model CNN. Pembagian data latih dan data validasi yang digunakan dapat dilihat pada Tabel 1.

Berdasarkan Tabel 1, data yang digunakan sebanyak 470 data citra yang terdiri dari 235 citra mobil dan 235 citra motor.

Contoh citra yang sudah melalui proses augmentasi dapat dilihat pada Gambar 2. Data yang ditampilkan hanya 9 dari keseluruhan gambar yang terdiri dari 4 motor dan 5 mobil. Pada proses ini juga dilakukan pemberian label pada data latih dan data validasi dengan menggunakan angka 0 dan 1. Angka 0 (nol) untuk label mobil dan angka 1 untuk label motor.

Tabel 1. Pembagian Data Penelitian

No	Jenis Kendaraan	Jumlah Data Latih	Jumlah Data Validasi
1	Mobil	188	47
2	Motor	188	47
3	Jumlah	376	94

Setelah dilakukan data augmentasi kemudian dilakukan proses segmentasi citra yang dilakukan dengan menggunakan algoritma k-means. Proses segmentasi dilakukan dengan menggunakan 3 *cluster* agar bentuk

objek dapat lebih mudah untuk dikenali berdasarkan warna untuk kemudian dilakukan pelatihan model. Contoh citra hasil segmentasi seperti pada Gambar 3. Gambar 3 menampilkan 12 citra hasil segmentasi.



Gambar 3. Hasil Data Augmentasi



Gambar 4. Hasil Segmentasi Citra

Pembuatan Model

Pada pembuatan model CNN, terdapat beberapa jenis lapisan yang digunakan, seperti lapisan *Conv2D* (*Convolutional 2D*), lapisan *AveragePooling2D*, lapisan *dropout*, lapisan *flatten*, dan lapisan *dense*. Jenis lapisan yang digunakan telah disesuaikan dengan kebutuhan peneliti. Pada proses konvolusi ini, terdapat 3 lapisan konvolusi *Conv2D*, 3 lapisan *pooling AveragePooling2D*, 3 lapisan *dropout*, 1 lapisan *flatten*, dan 4 lapisan *dense*. Masing-masing lapisan memiliki peran dan fungsi tertentu dalam proses ekstraksi dan klasifikasi fitur pada data gambar.

Pertama, terdapat lapisan konvolusi *Conv2D* dengan 256 *filter* dan kernel 3x3, menghasilkan *output* dengan ukuran 126x126. Lapisan ini bertanggung jawab dalam mengekstrak fitur dari gambar *input*. Setelah itu, dilakukan lapisan *pooling AveragePooling2D* dengan *filter* 2x2, yang memperkecil ukuran *output*

menjadi 63x63. Proses *pooling* ini membantu mengurangi dimensi data dan mempertahankan fitur-fitur penting. Lapisan *dropout* kemudian digunakan untuk mengurangi *overfitting* secara acak mengabaikan sebagian unit pada lapisan sebelumnya. Dalam model ini, tingkat *dropout* yang digunakan adalah 0.2.

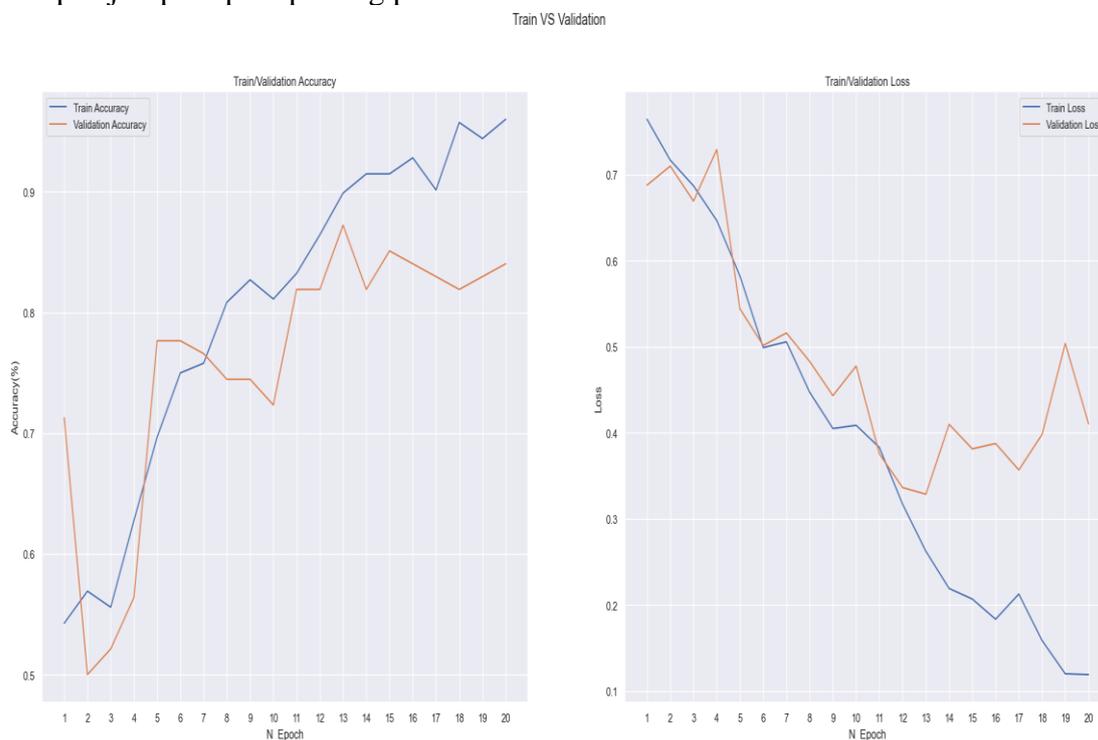
Proses konvolusi dilakukan kembali dengan lapisan *Conv2D* kedua, menggunakan 128 *filter* dan kernel 3x3. *Outputnya* memiliki ukuran 61x61. Kemudian diikuti dengan lapisan *pooling AveragePooling2D* kedua, yang memperkecil ukuran *output* menjadi 30x30. Lapisan *dropout* kedua digunakan lagi dengan tingkat *dropout* 0.2. Setelah itu, dilakukan konvolusi lagi dengan lapisan *Conv2D* ketiga yang memiliki 64 *filter* dan kernel 3x3. *Outputnya* memiliki ukuran 28x28, diikuti dengan lapisan *pooling AveragePooling2D* ketiga yang memperkecil ukuran *output* menjadi 14x14. Kembali dilakukan lapisan *dropout* dengan tingkat *dropout* 0.2.

Lalu, *output* dari lapisan sebelumnya diubah menjadi bentuk vektor dengan lapisan *flatten*, sehingga memiliki dimensi (*None*, 12,544). Vektor ini akan menjadi *input* untuk lapisan-lapisan *dense* berikutnya. Dua lapisan *dense* terakhir digunakan untuk melakukan klasifikasi. Lapisan *dense* pertama memiliki 512 unit dengan fungsi aktivasi ReLU, dan lapisan *dense* kedua memiliki 256 unit. Lapisan *dense* terakhir memiliki 1 unit dengan fungsi aktivasi *sigmoid*, yang digunakan untuk menghasilkan probabilitas klasifikasi.

Dengan menggunakan kombinasi lapisan konvolusi, *pooling*, *dropout*, *flatten*, dan *dense*, model CNN ini dapat mempelajari pola-pola penting pada data

gambar dan melakukan klasifikasi dengan akurasi yang tinggi. Model ini dirancang dengan lebih banyak lapisan untuk melihat bagaimana kinerjanya dan sejauh mana model ini dapat mempelajari fitur-fitur penting dalam data gambar.

Berikutnya yaitu melakukan pelatihan data citra mobil dan motor dengan melakukan *fit model*. *Fit model* dilakukan dengan *epoch* 20. Epoch digunakan sebagai perulangan bagi model yang dibangun untuk mengenali semua data latih dan data validasi yang sudah dilakukan pemrosesan dengan data *augmentasi* sebelumnya. Hasil *loss* dan *accuracy* model dapat dilihat pada Gambar 5.



Gambar 5. Grafik iterasi nilai *loss* dan *accuracy*

Pada tabel 2 dapat dilihat nilai *loss* pada iterasi ke 20 pada data latih yaitu 0.119171 dan *accuracy* pada data latih yaitu 0.960106. Kemudian pada data validasi menghasilkan nilai *loss* 0.410513 dengan nilai *accuracy* sebesar 0.840426. Berdasarkan hasil pengujian dapat diperoleh informasi bahwa dengan

semakin banyak perulangan maka nilai *loss* semakin berkurang dan *accuracy* semakin meningkat. Tetapi terlalu banyak nilai *epoch* juga dapat memberikan dampak lain yaitu terjadinya *overfitting*. Hasil dari rekapitulasi grafik pada Gambar 5 dapat dilihat pada Tabel 2.

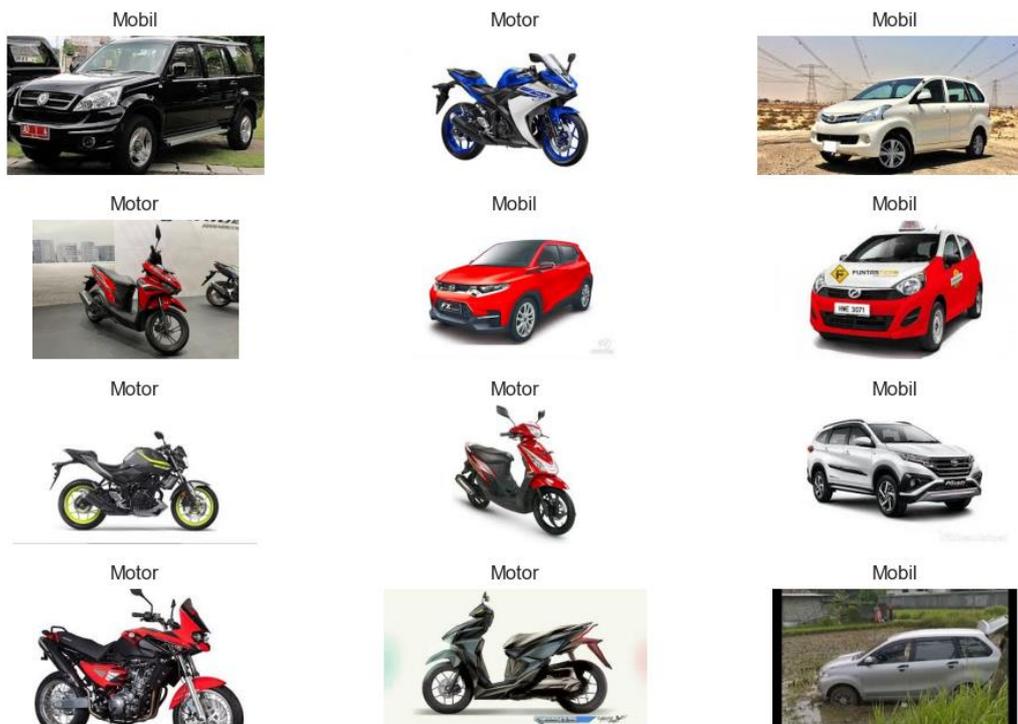
Tabel 2. Nilai Hasil Iterasi Nilai *Accuracy* dan *Loss*

<i>Epoch</i>	<i>Train loss</i>	<i>Train accuracy</i>	<i>Validation loss</i>	<i>Validation accuracy</i>
1	0.764159	0.542553	0.687665	0.712766
2	0.716732	0.569149	0.709750	0.500000
3	0.686903	0.555851	0.669153	0.521277
4	0.646656	0.627660	0.728936	0.563830
5	0.582084	0.696809	0.543873	0.776596
6	0.499023	0.750000	0.501523	0.776596
7	0.505724	0.757979	0.515950	0.765957
8	0.447164	0.808511	0.482918	0.744681
9	0.405044	0.827128	0.443185	0.744681
10	0.408737	0.811170	0.477599	0.723404
11	0.383009	0.832447	0.375491	0.819149
12	0.317080	0.864362	0.336413	0.819149
13	0.262491	0.898936	0.328697	0.872340
14	0.219268	0.914894	0.409873	0.819149
15	0.206909	0.914894	0.381374	0.851064
16	0.183495	0.928191	0.387593	0.840426
17	0.212760	0.901596	0.356862	0.829787
18	0.158791	0.957447	0.398127	0.819149
19	0.120100	0.944149	0.503720	0.829787
20	0.119171	0.960106	0.410513	0.840426

Uji Coba dan Evaluasi

Hasil pengujian pada data uji yang terdiri dari 12 citra yang terbagi menjadi 6 mobil dan juga 6 motor dapat dilihat pada gambar 6. Berdasarkan pengujian model yang dilakukan terhadap data uji,

6 motor berhasil diklasifikasi dengan benar dan 6 mobil berhasil diklasifikasi dengan benar. Dengan demikian model mampu memprediksi dengan baik yaitu akurasi 100% secara keseluruhan dari data uji yang sudah disiapkan.



Gambar 6. Hasil Klasifikasi (Prediksi)

KESIMPULAN

Berdasarkan implementasi pengujian model dan klasifikasi citra menggunakan metode *convolutional neural network* (CNN) pada mobil dan motor dengan *library* keras telah menghasilkan akurasi sebesar 100%. Dimana dengan tahapan pra-pemrosesan data menggunakan data augmentasi untuk memperluas dataset yang kemudian dilakukan pemisahan data latih dan data validasi. Selanjutnya, hasil data latih dilakukan segmentasi dengan menggunakan 3 *clusters*. Dari hasil segmentasi dilakukan pemodelan data menggunakan lapisan konvolusi *Conv2D* dengan 256 *filter* dengan kernel 3x3, kemudian lapisan *pooling AveragePooling2D* dengan *filter* 2x2, terakhir dilakukan *dropout*, *flatten*, dan *dense*, yang dirancang untuk melihat bagaimana kinerjanya serta sejauh mana model ini dapat mempelajari fitur-fitur penting dalam data gambar. Berikutnya melakukan pelatihan data citra mobil dan motor dengan melakukan *fit model*

sebanyak *epoch* 20. Hasil *loss* dan *accuracy* model dapat dilihat pada gambar 5 yang dapat dilihat nilai *loss* pada iterasi ke 20 pada data latih yaitu 0.119171 dan *accuracy* pada data latih yaitu 0.960106. Kemudian pada data uji menghasilkan nilai *loss* 0.410513 dengan nilai *accuracy* sebesar 0.840426. Pada penelitian selanjutnya untuk menurunkan nilai *loss* dan meningkatkan nilai *accuracy* dapat menambahkan beberapa data citra, menambah lapisan konvolusi dan lapisan *pooling*, serta melakukan segmentasi dengan kombinasi metode lain sehingga diperoleh akurasi klasifikasi yang lebih baik dengan data uji yang lebih banyak.

DAFTAR PUSTAKA

Abdulhakim, R., & Dermawan, B. A. (2021). Analisis dan Penerapan Algoritma Convolutional Neural Network untuk Klasifikasi Kendaraan Prioritas. *Jurnal Sains Dan Informatika*, 7(2), 135–144.

- Annur, H. (2018). Klasifikasi Masyarakat Miskin Menggunakan Metode Naive Bayes. *ILKOM Jurnal Ilmiah*, 10(2), 160–165.
- Destyningtias B., Heranurweni S., & T. Nurhayati. (2010). Segmentasi Citra Dengan Metode Pengembangan. *Jurnal ElektriKa*, 2(1), 39–49.
- Fonda, H. (2020). Klasifikasi Batik Riau Dengan Menggunakan Convolutional Neural Networks (Cnn): Klasifikasi Batik Riau Dengan Menggunakan Convolutional Neural Networks (Cnn). *Jurnal Ilmu Komputer*, 9(1), 7–10.
- Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia. *Algor*, 2(1), 12–20.
- Primartha, R. (2018). Belajar Machine Learning Teori dan Praktik. Bandung: Informatika Bandung, 10, 20–30.
- Putra, W. S. E. (2016). Klasifikasi citra menggunakan convolutional neural network (CNN) pada caltech 101. *Jurnal Teknik ITS*, 5(1).
- Santoso, A., & Gunawan Ariyanto, S. (2018). Implementasi deep learning berbasis keras untuk pengenalan wajah. Universitas Muhammadiyah Surakarta.
- Sulistiyawati, A., & Supriyanto, E. (2021). Implementasi Algoritma K-means Clustering dalam Penentuan Siswa Kelas Unggulan. *Jurnal Tekno Kompak*, 15(2), 25–36.