

# KLASIFIKASI CUACA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

## WEATHER CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD

Randhy Sulisty Budi<sup>1</sup>, Raditiana Patmasari<sup>2</sup>, Sofia Saidah<sup>3</sup>

<sup>1,2,3</sup>Universitas Telkom, Bandung

randhysb@students.telkomuniversity.ac.id<sup>1</sup>, raditiana@telkomuniversity.ac.id<sup>2</sup>, sofiasaidahsfi@telkomuniversity.ac.id<sup>3</sup>

### Abstrak

Cuaca merupakan kondisi pada atmosfer yang terjadi karena berbagai faktor, antara lain radiasi matahari, suhu udara, kecepatan dan arah angin. Aktivitas kehidupan manusia sangat bergantung pada faktor cuaca terutama pada sektor pertanian, penerbangan, dan lain-lain. Memahami kondisi cuaca menjadi hal penting untuk kehidupan manusia, sehingga diperlukan sebuah sistem untuk mengklasifikasikan cuaca sesuai dengan kategorinya agar informasi yang dihasilkan dapat digunakan untuk berbagai kebutuhan. Metode yang diajukan yaitu menggunakan *Deep Learning* dengan metode *Convolutional Neural Network (CNN)*. *Convolutional Neural Network (CNN)* merupakan salah satu metode *Deep Learning* yang berfungsi untuk mengolah data dua dimensi, yang dapat diimplementasikan menjadi data citra. *Dataset* citra yang digunakan terdiri dari tiga kelas citra cuaca yaitu berawan, hujan, dan cerah. Arsitektur VGG16net digunakan pada penelitian ini karena lapisan konvolusinya yang sederhana, yaitu hanya menggunakan lapisan konvolusi 3x3. Proses klasifikasi cuaca CNN terdiri dari tahap *training* dan *testing* untuk mengetahui hasil klasifikasinya. Parameter kinerja yang dianalisis berupa akurasi, presisi, *recall*, dan *F1-Score*. Parameter *optimizer*, *epoch* dan *learning rate* dapat mempengaruhi hasil performa sistem berupa nilai akurasi, presisi, *recall*, dan *F1-score*. Pada penelitian ini menghasilkan performa terbaik yaitu dengan penggunaan *optimizer RMSprop*, resolusi citra 224x224, *learning rate* 0,0001, dan *epoch* 50 dengan nilai performa sistem untuk akurasi, presisi, *recall*, dan *F1-score* masing-masing sebesar 87%, 90%, 87%, dan 87%.

**Kata Kunci** : Cuaca, *Deep Learning*, *Convolutional Neural Network*, *Machine Learning*

### Abstract

*Weather is a condition in the atmosphere that occurs due to various factors, including solar radiation, air temperature, wind speed and direction. Human life activities are highly dependent on weather factors, especially in the agricultural sector, aviation, and others. Understanding weather conditions is important for human life, so we need a system to classify weather according to its category so that the resulting information can be used for various needs. The proposed method is using Deep Learning with the Convolutional Neural Network (CNN) method. Convolutional Neural Network (CNN) is a Deep Learning method that functions to process two-dimensional data, which can be implemented into image data. The image dataset used consists of three classes of weather images, namely cloudy, rainy, and sunny. The VGG16net architecture is used in this research because of its simple convolution layer, which only uses a 3x3 convolution layer. The CNN weather classification process consists of training and testing stages to determine the classification results. The performance parameters analyzed were accuracy, precision, recall, and F1-Score. The optimizer, epoch and learning rate parameters can affect system performance results in the form of accuracy, precision, recall, and F1-score values. In this study, the best performance was the use of the RMSprop optimizer, 224x224 image resolution, learning rate 0.0001, and epoch 50 with system performance values for accuracy, precision, recall, and F1-score respectively 87%, 90%, 87%, and 87%.*

**Keywords:** *Weather, Convolutional Neural Network, Machine Learning*

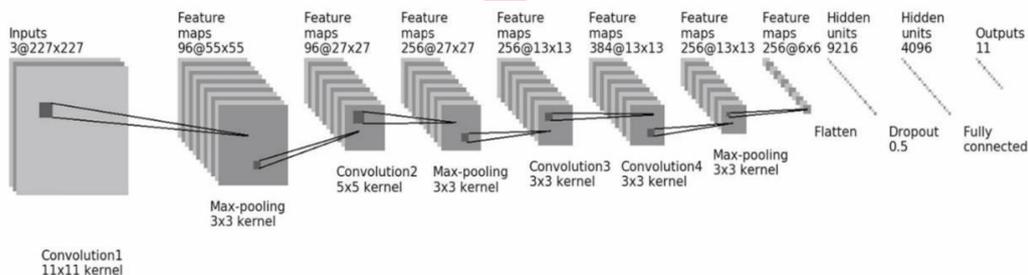
1. Pendahuluan

Cuaca merupakan kondisi pada atmosfer yang terjadi karena berbagai faktor, antara lain radiasi matahari, suhu udara, kecepatan dan arah angin[1]. Jenis-jenis cuaca meliputi cuaca cerah, berawan, panas, dingin, berangin, dan hujan. Aktivitas kehidupan manusia sangat bergantung pada kondisi cuaca. Pada bidang pertanian, musim hujan yang intens dapat membatasi pertumbuhan tanaman, seperti pertumbuhan tanaman jagung pada musim hujan[2]. Pengaplikasian klasifikasi dari unsur-unsur cuaca dapat menjadi pedoman keadaan cuaca pada suatu tempat. Klasifikasi unsur-unsur cuaca sering digunakan dalam keperluan peramalan cuaca[3], dan pedoman *deep learning* dalam mengambil keputusan. Dalam pemanfaatannya, sistem komputer dapat mengetahui keadaan cuaca dan menilai keadaan cuaca sesuai dengan cuaca yang terjadi. Oleh karena itu diperlukan sebuah sistem untuk mengklasifikasikan cuaca sesuai dengan kategorinya agar dapat memberikan hasil yang lebih akurat, dan mengurangi kesalahan dari pengamat cuaca.

Pada tahun 2018, penelitian klasifikasi curah hujan dilakukan menggunakan metode Naïve Bayesian. Penelitian ini menggunakan Naïve Bayesian untuk menentukan probabilitas suatu keanggotaan dari suatu class dengan data set BMKG di wilayah bali dengan data yang mendukung kategori curah hujan. Terdapat 5 *feature* dalam menentukan klasifikasi curah hujan yaitu suhu minimum, suhu maksimum, kelembaban rata-rata, lama penyiraman, dan kecepatan angin[4]. Penelitian tentang sistem klasifikasi cuaca menggunakan *Convolutional Neural Network* (CNN) dikombinasikan dengan *multiclass Support Vector Machine* (SVM) telah dilakukan pada tahun 2018 untuk mendapatkan hasil klasifikasi cuaca dengan *dataset* penelitian sebelumnya untuk cuaca cerah dan berawan, *desnownet* untuk cuaca bersalju, dan *dataset d-hazy*[5]. Pada penelitian yang akan dilakukan terdapat perbedaan yaitu menggunakan Kaggle sebagai *dataset* yang akan digunakan. Kaggle merupakan anak perusahaan Google LLC, yaitu komunitas *online* untuk praktisi *machine learning* dan *data scientist*. Metode yang digunakan pada penelitian ini adalah *Convolutional Neural Network* (CNN) untuk klasifikasi cuaca, dengan menggunakan arsitektur VGG16Net.

2. Convolution Neural Network (CNN)

*Convolutional Neural Network* (CNN) berfungsi untuk memproses data dalam bentuk *multiple array*. Terdapat tiga layer atau lapisan pada CNN, yaitu *Convolutional Layer*, *Pooling Layer*, dan *Fully Connected Layer*. Ilustrasi arsitektur CNN ditunjukkan pada Gambar 1.

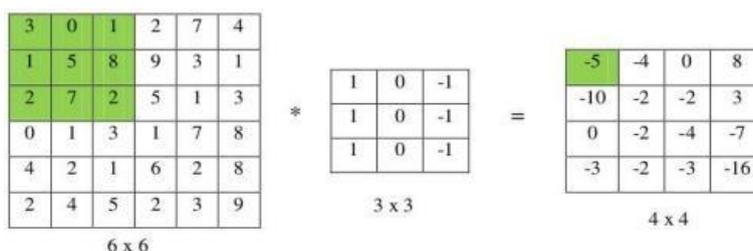


Gambar 1. Arsitektur CNN

Pada *Convolutional Layer*, representasi baru dari citra diperoleh dengan menggerakkan *convolution kernel* atau filter dengan ukuran tertentu. Representasi terbaru dihasilkan dari perkalian nilai piksel citra terhadap filter yang digunakan. *Pooling layer* berfungsi untuk mengecilkan ukuran *activation map* dengan *downsampling* dan setiap *activation map* dilakukan *downsampling* terpisah. Nilai *pooling* umumnya diambil dari yang terbesar atau biasa disebut dengan *max-pooling*, bisa juga menggunakan *average-pooling*. *Fully Connected Layer* digunakan sebagai klasifier yang berfungsi untuk menerima input dalam bentuk vector *feature map*. Hasil *feature map* dari layer sebelumnya perlu dibentuk ulang karena masih dalam bentuk multidimensional array, pembentukan ulang atau disebut dengan *flatten* pada *feature map* dilakukan agar mendapatkan input untuk *fully connected layer*. Pada CNN, *convolutional layer* dan *pooling layer* berfungsi sebagai fitur ekstraksi, sedangkan *fully connected layer* berfungsi sebagai fitur klasifier. Terdapat beberapa jenis arsitektur pada CNN, salah satunya yaitu VGG16net. Arsitektur VGG16net digunakan pada penelitian kali ini karena lapisan konvolusinya yang sederhana, yaitu hanya menggunakan lapisan konvolusi 3x3.

2.1 Convolution Layer

*Convolution layer* merupakan lapisan pertama pada CNN. Lapisan tersebut akan melakukan kovolusi citra masukan dengan *filter* yang telah didefinisikan tanpa merusak struktur citra awal. Fungsi *convolution layer* yaitu mengambil fitur pada citra yang akan digunakan untuk melatih model. Gambar contoh *convolutional layer* ditunjukkan pada Gambar 2.



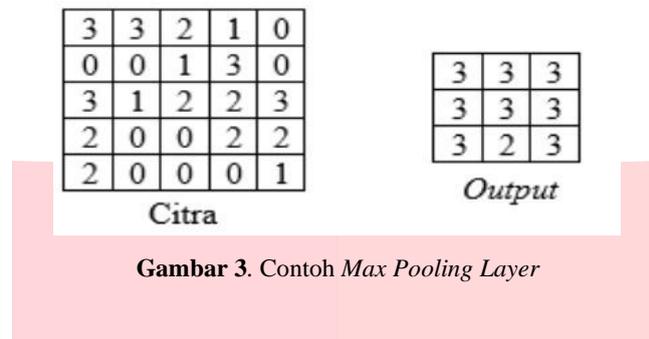
Gambar 2. Contoh Convolutional Layer

2.2 Pooling Layer

Pooling layer yaitu proses mengurangi jumlah parameter dan jumlah perhitungan pada jaringan, serta melakukan pencegahan *overfitting* pada citra. Layer ini terbagi menjadi *average pooling* dan *max pooling*. *Average pooling* yaitu mengambil nilai rata-rata dari area yang dipilih, sedangkan *max pooling* yaitu mengambil nilai terbesar dari area yang dipilih. Persamaan dari *max pooling* ditunjukkan pada Persamaan 1.

$$a_j = \underset{N \times N}{\text{Max}} \underset{i}{a^{n \times n}} \mu(n, n) \tag{1}$$

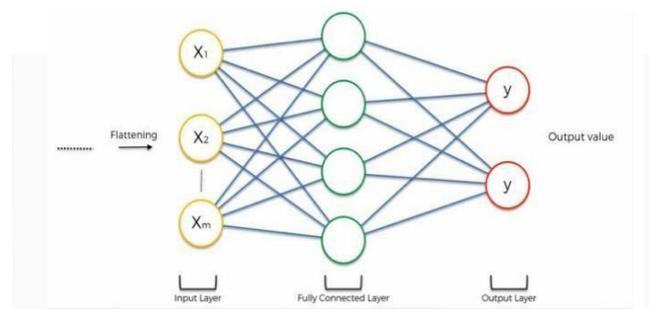
$a_i$  adalah nilai dari *pooling map*,  $a_j$  adalah nilai dari *input map*, dan  $\mu(n, n)$  adalah *window function*. Contoh *max pooling layer* ditunjukkan pada Gambar 3.



Gambar 3. Contoh Max Pooling Layer

2.3 Fully Connected Layer

Fully connected layer merupakan lapisan dimana dimensi seluruh data akan diubah menjadi satu dimensi. Proses perubahan ukuran dimensi disebut *flatten*. Lapisan ini memiliki *node* yang saling berhubungan, memiliki bobot, dan memiliki fungsi aktivasi. Lapisan ini mengeluarkan sebuah prediksi berdasarkan data masukan. Ilustrasi *fully connected layer* dapat dilihat pada Gambar 4.



Gambar 4. Ilustrasi Proses Fully Connected Layer

2.4 Rectified Linear Units (RELU)

Rectified Linear Unit (RELU) adalah fungsi aktivasi yang memiliki kelebihan yaitu mampu memproses data ukuran besar dengan cepat, yang digunakan di antara *convolutional layer* dan *pooling layer*. RELU menjaga hasil citra konvolusi pada domain definit positif, sehingga setiap nilai negatif yang berasal dari proses konvolusi akan melalui proses RELU, dan menjadikan nilai negatif tersebut sama dengan 0. Persamaan RELU ditunjukkan pada Persamaan 2.

$$f = \max(0, x) \tag{2}$$

2.5 Softmax

Softmax adalah fungsi aktivasi yang sering digunakan pada *neural network* yang memiliki banyak kategori output (*multi-class*). Fungsi *softmax* mengubah nilai perhitungan menjadi nilai *probability*, hal ini menjadikan nilai hasil perhitungan bisa dibandingkan. Dengan menggunakan fungsi *softmax* dapat terlihat kelas mana yang memiliki nilai kemungkinan terbesar. Kemudian, kemungkinan terbesar tersebut akan menjadi kelas terpilih dan masukan selanjutnya akan terklasifikasi menjadi kelas tersebut. Persamaan fungsi *softmax* ditunjukkan pada Persamaan 3.

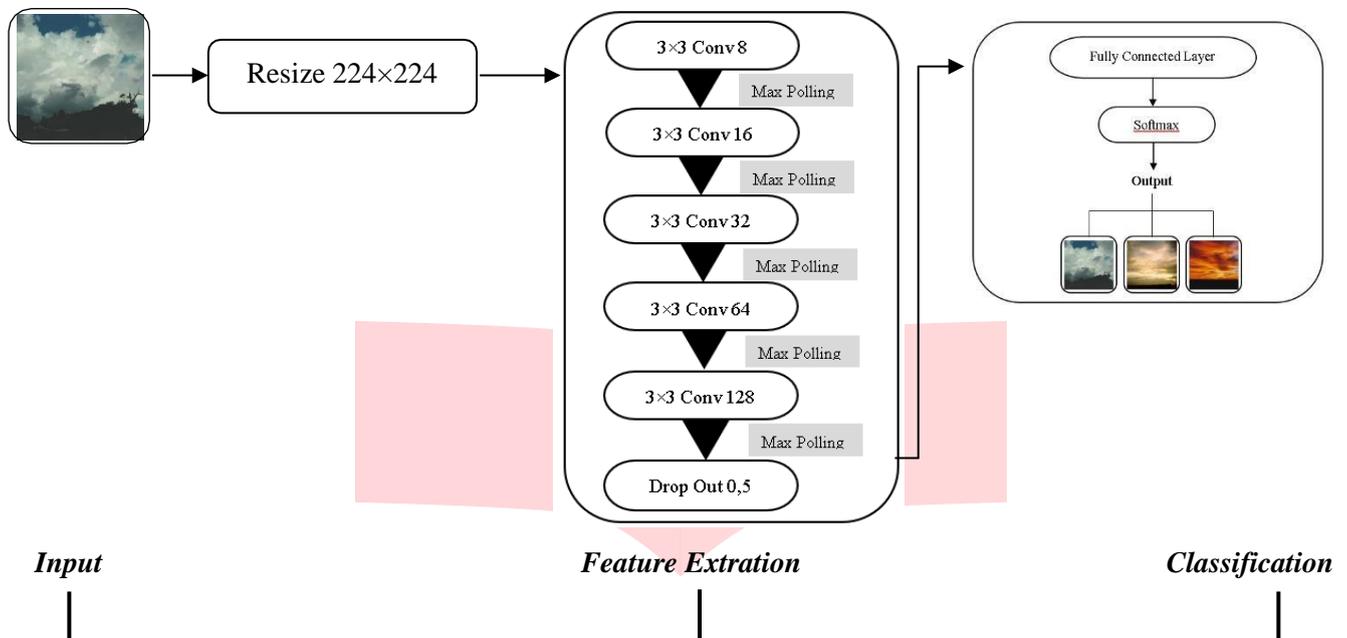
$$f_i(x) = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}} \tag{3}$$

where  $i = 1, 2, 3, \dots, k$   
 $0 \leq f_i(x) \leq 1$

dimana k adalah jumlah input, dan x adalah vektor input.

### 3. Desain Sistem

Gambaran umum sistem yang akan digunakan pada penelitian ini untuk proses klasifikasi cuaca ditunjukkan pada Gambar 5.



Gambar 5. Model Sistem CNN Usulan

Berdasarkan Gambar 5 model sistem CNN usulan tersebut akan digunakan pada proses klasifikasi cuaca. Pada proses *input* terdapat tiga kelas citra cuaca yang terdiri dari berawan, hujan, dan cerah, dengan data latih sejumlah 75% dan data uji 25%. Ukuran citra cuaca yang tidak sama rata akan disamakan ukurannya menjadi 224x224 *pixel*. Hasil scan data citra terdiri dari *channel Grayscale*.

Proses selanjutnya *input* citra masuk ke dalam proses *feature extraction learning*. Pada proses tersebut akan dilakukan proses konvolusi, aktivasi ReLU, dan *pooling*. Pada proses *feature learning* terdapat 5 lapisan konvolusi, 3 *fully connected layer*, dan 1 aktivasi *softmax*. Ukuran yang sama akan digunakan pada setiap lapisan konvolusi yaitu ukuran kernel 3x3. Jumlah kanal *output* pada setiap *hidden layer* masing-masing sebesar 8, 16, 32, 64, 128. Pada proses konvolusi setiap *layer* menggunakan *activation ReLU* akan dilakukan proses *max pooling*. Hasil *max pooling* berupa citra yang telah direduksi.

Proses yang dilakukan jika *max pooling* selesai yaitu proses *classification*. Proses *classification* dimulai dengan proses perampingan atau *flattening*. Proses yang terjadi pada *flattening* yaitu mengubah *feature map* hasil *pooling layer* menjadi sebuah *vector* atau 1 dimensi, tahap ini juga dapat disebut *fully connected layer*. Proses terakhir yang dilakukan untuk klasifikasi data citra menjadi tiga kelas *output* cuaca yaitu fungsi aktivasi *softmax*.

### 4. Performansi Sistem

Jika seluruh tahap pada *training* data dan *testing* data sudah dilakukan, langkah selanjutnya yaitu dilakukan performansi sistem. Parameter untuk menentukan performansi sistem terdiri dari akurasi, presisi, *recall*, dan *f1-score*. Proses pengukuran parameter tersebut dibantu menggunakan *Confusion Matriks*. *Confusion matriks* merupakan alat pengukuran yang berfungsi untuk menghitung tingkat kebenaran dari proses klasifikasi. Setiap kelas-kelas yang berbeda dapat dianalisis tingkat *classifiernya* menggunakan *Confusion Matriks*. Tabel *Confusion Matriks* ditunjukkan pada Tabel 1.

Tabel 1. Confusion Matriks

		Prediksi	
		Positif	Negatif
Data Uji	Positif	TP	FN
	Negatif	FP	TN

Keterangan:

- TP (*True Positive*) merupakan data positif yang di prediksi benar (positif).
- TN (*True Negative*) merupakan data negatif yang diprediksi benar (negatif).
- FP (*False Positive*) merupakan data negatif namun diprediksi sebagai data positif.
- FN (*False Negative*) merupakan data positif namun diprediksi sebagai data negatif.

Nilai akurasi menggambarkan rasio prediksi benar (positif dan negatif) berdasarkan keseluruhan data. Rentang hasil dari parameter akurasi berkisar 0 sampai 1. Hasil akurasi kemudian dapat diubah menjadi presentase. Semakin tinggi presentase akurasi maka sebuah algoritma yang digunakan pada sistem semakin baik. Persamaan akurasi ditunjukkan pada Persamaan 4.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Presisi merupakan proporsi prediksi positif yang benar dengan seluruh prediksi positif. Presisi menggambarkan tingkat ketepatan sebuah model dalam memprediksi kejadian positif pada sebuah rangkaian kegiatan prediksi. Persamaan presisi ditunjukkan pada Persamaan 5.

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (5)$$

*Recall* atau dikenal juga dengan istilah *True Positive* (TP) digunakan untuk menggambarkan keberhasilan jenis citra mikroskopis yang benar saat melakukan identifikasi. Persamaan *recall* ditunjukkan pada Persamaan 6.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

*F1-score* menggambarkan gabungan nilai *recall* dan *precision* ke dalam variabel nilai baru, karena terdapat kecenderungan saat nilai *recall* tinggi maka nilai presisi menjadi rendah dan begitu pula sebaliknya. Persamaan *F1-score* ditunjukkan pada Persamaan 7.

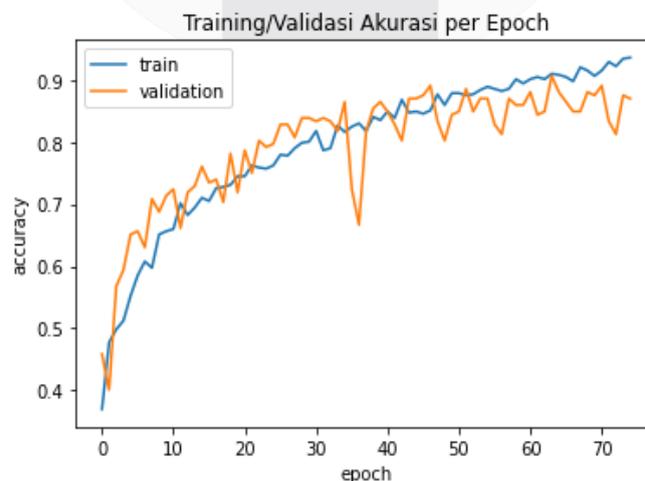
$$F1score = 2 \times \frac{\text{Recall} \times \text{Presisi}}{\text{Recall} + \text{Presisi}} \quad (7)$$

## 5. Hasil dan Diskusi

Pada penelitian ini akan fokus membahas pengujian hasil dan analisis dari klasifikasi cuaca dengan arsitektur VGG16Net. *Dataset* yang digunakan berisi 768 citra dengan format JPG yang terdiri dari tiga kelas cuaca yaitu berawan, hujan, dan cerah. Pada proses pengolahan data *training* peneliti mengambil sejumlah 75% citra, dan 25% digunakan sebagai data validasi. Hasil pengujian klasifikasi cuaca terdiri dari akurasi, presisi, *f1-score* dan *recall*.

Terdapat empat pengujian yang akan dilakukan pada tahap perancangan pengujian. Skenario pertama bertujuan untuk mendapatkan *optimizer* terbaik dengan parameter yang ditentukan. Skenario ke dua untuk mendapatkan ukuran citra terbaik, dengan *optimizer* terbaik yang telah ditentukan pada skenario sebelumnya. Skenario ke tiga untuk mendapatkan *learning rate* terbaik dengan menggunakan *optimizer* dan ukuran citra terbaik berdasarkan skenario sebelumnya. Skenario terakhir untuk mendapatkan *epoch* terbaik dengan menggunakan *optimizer*, ukuran citra, dan *learning rate* terbaik yang telah ditemukan pada dua skenario sebelumnya.

Parameter terbaik yang didapatkan dari empat pengujian yaitu menggunakan *optimizer* RMSprop, dengan ukuran citra  $224 \times 224$ , *learning rate* 0,0001, dan *epoch* 75, dengan nilai performansi sistem akurasi, presisi, *recall*, dan *f1-score* masing-masing sebesar 87%, 90%, 87%, dan 87%. Grafik hasil pengujian tidak mengalami *overfitting* karena jarak hasil *training* dan *testing* pada grafik berdekatan. Grafik hasil ditunjukkan pada Gambar 6.



Gambar 6. Grafik Hasil Akurasi per Epoch

## 6. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan terhadap Klasifikasi Cuaca Menggunakan *Convolutional Neural Network* (CNN) disimpulkan bahwa mampu mengklasifikasi jenis cuaca berdasarkan kelasnya. Pengujian menggunakan metode CNN dalam proses klasifikasi cuaca dengan mengubah parameter jenis *optimizer*, ukuran citra, *learning rate*, dan *epoch* sangat berpengaruh terhadap tingkat akurasi, presisi, *recall* dan *F1-score* sistem. Pengujian terbaik atau parameter terbaik didapatkan dari pengambilan hasil parameter terbaik yang telah diuji pada skenario pertama sampai scenario ke empat. Parameter terbaik didapatkan berdasarkan penggunaan *optimizer* RMSprop, dengan ukuran citra 224x224, *Epoch* 75, dan *learning rate* 0,0001 dengan nilai performa sistem akurasi, presisi, *recall*, dan *F1-score* masing-masing sebesar 87%, 90%, 87%, dan 87%. Hasil pengujian tidak mengalami *overfitting* karena jarak hasil *training* dan *testing* pada grafik berdekatan. Berdasarkan hasil dari performa sistem yang diperoleh dapat disimpulkan model yang diusulkan untuk sistem klasifikasi cuaca tidak mengalami *overfitting* maupun *overshooting* atau sistem mampu mengenali kondisi dari data citra latih dan data citra validasi dengan akurasi sebesar 87%.

## Referensi

- [1] W. Suryanto and A. Luthfian, "PENGANTAR METEOROLOGI," Yogyakarta, UGM PRESS, 2019, p. 10.
- [2] S. Gischa, "Kompas.com," Kompas.com, 2021. [Online]. Available: <https://www.kompas.com/skola/read/2021/03/01/191900169/kondisi-cuaca-dan-pengaruhnya-terhadap-kegiatan-manusia?page=all> . [Accessed 22 March 2021].
- [3] S. Ernawati, "APLIKASI HOPFIELD NEURAL NETWORK UNTUK PRAKIRAAN CUACA," *Jurnal Meteorologi dan Geofisika*, vol. 10, no. 2, 2009.
- [4] I. G. A. Gunadi and A. Aprilyana Kusuma Dewi, "Klasifikasi Curah Hujan di Provinsi Bali Berdasarkan Metode Naïve Bayesian," *Wahana Matematika dan Sains: Jurnal Matematika, Sains, dan Pembelajarannya*, vol. 12, no. 1, 2018.
- [5] J. An, Y. Chen and H. Shin, "Weather Classification using Convolutional Neural Networks," in *International SoC Design Conference (ISOCC)*, Daegu, 2018.