



# Laporan Final Project

## Kecerdasan Buatan (Lanjut)

*Deteksi Jenis Cuaca dari Citra Langit Menggunakan CNN*



Kelompok 4

Anggota Kelompok:

1. 2303010121 | Muhamad Riyam Faujan | Universitas Perjuangan Tasikmalaya
2. 2303010127 | Alfiyah Adwa | Universitas Perjuangan Tasikmalaya
3. 23.11.5685 | Yezha amanando rifky zafian | Universitas Amikom Yogyakarta
4. 23.11.5685 | Jerico Rosales Siahay | Universitas Amikom Yogyakarta

## **1. Latar Belakang**

Cuaca merupakan kondisi alam yang sangat memengaruhi kehidupan sehari-hari manusia, baik dalam aktivitas transportasi, pertanian, pariwisata, maupun aspek keselamatan. Oleh karena itu, informasi mengenai cuaca yang tepat dan dapat diandalkan menjadi kebutuhan penting bagi masyarakat serta lembaga terkait dalam menentukan langkah dan kebijakan. Selama ini, pemantauan cuaca umumnya dilakukan melalui perangkat sensor meteorologi yang mengukur suhu, kelembapan, tekanan udara, dan curah hujan. Namun, penggunaan alat tersebut memerlukan sarana khusus dan belum tentu tersedia secara merata di semua daerah.

Seiring dengan perkembangan teknologi dan semakin mudahnya akses terhadap perangkat kamera, seperti kamera digital, drone, dan ponsel pintar, citra langit dapat dimanfaatkan sebagai sumber informasi alternatif untuk mengenali kondisi cuaca. Tampilan visual langit, seperti perubahan warna, intensitas cahaya, keberadaan awan, serta pola atmosfer, dapat memberikan gambaran mengenai kondisi cuaca tertentu, misalnya cerah, berawan, hujan, atau waktu matahari terbit. Meski demikian, penentuan cuaca berdasarkan pengamatan visual secara manual masih bersifat subjektif dan kurang efektif jika diterapkan dalam jumlah data yang besar.

Kemajuan di bidang kecerdasan buatan, khususnya teknologi Deep Learning, memungkinkan proses analisis citra dilakukan secara otomatis dengan hasil yang lebih konsisten dan akurat. Salah satu metode yang banyak digunakan adalah Convolutional Neural Network (CNN), yang mampu mengenali dan mengekstraksi ciri visual dari sebuah gambar dengan baik. Metode ini telah banyak diterapkan pada berbagai bidang, seperti pengenalan wajah, klasifikasi objek, dan analisis citra medis, sehingga memiliki potensi besar untuk diterapkan dalam pengenalan kondisi cuaca berbasis citra langit.

Berdasarkan hal tersebut, penelitian ini bertujuan untuk mengembangkan model Convolutional Neural Network yang dapat mendeteksi dan mengelompokkan jenis cuaca berdasarkan citra langit. Dengan memanfaatkan dataset citra cuaca yang tersedia secara terbuka, diharapkan sistem yang dibangun mampu menghasilkan klasifikasi yang akurat dan dapat digunakan sebagai pendukung dalam sistem pemantauan cuaca berbasis visual. Selain itu, hasil penelitian ini diharapkan dapat memberikan kontribusi dan menjadi acuan bagi pengembangan aplikasi cerdas di bidang lingkungan dan meteorologi berbasis pengolahan citra digital.

## **2. Dataset**

Dataset yang digunakan dalam penelitian ini merupakan dataset publik yang diperoleh dari platform Kaggle dengan nama Multi-class Weather Dataset. Dataset ini berisi kumpulan citra langit yang telah diberi label berdasarkan kondisi cuaca tertentu dan dapat diakses melalui tautan berikut:

<https://www.kaggle.com/datasets/pratik2901/multiclass-weather-dataset>

Dataset terdiri dari beberapa kelas cuaca, yaitu cloudy (berawan), rain (hujan), shine (cerah), dan sunrise (matahari terbit). Seluruh citra berbentuk gambar berwarna (RGB) dengan variasi resolusi dan kondisi pencahayaan. Sebelum digunakan dalam proses pelatihan model, data dilakukan tahap pra-pemrosesan berupa penyesuaian ukuran gambar dan normalisasi nilai piksel. Dataset kemudian dibagi menjadi data latih dan data validasi untuk keperluan evaluasi model.

### 3. Metode:

- **Alur Pengerjaan**

Alur pengerjaan project deteksi jenis cuaca dari langit menggunakan Convolutional Neural Network (CNN) dilakukan melalui tahapan-tahapan berikut :

1. **Pengumpulan Dataset**

Mengumpulkan dataset citra langit dari sumber publik yang berisi berbagai kondisi cuaca dan telah memiliki label kelas.

2. **Preprocessing Data**

Melakukan penyesuaian ukuran citra, normalisasi nilai piksel, serta augmentasi data untuk meningkatkan kualitas dan variasi data.

3. **Pembagian Dataset**

Membagi dataset menjadi data latih, data validasi, dan data uji untuk keperluan pelatihan dan evaluasi model.

4. **Perancangan Model**

Merancang arsitektur Convolutional Neural Network yang terdirir dari lapisan konvolusi, pooling, dan fully connected.

5. **Pelatihan Model**

Melatih model CNN menggunakan data latih hingga diperoleh performa yang optimal.

6. **Pengujian Model**

Menguji model menggunakan data uji untuk mengetahui kemampuan klasifikasi jenis cuaca.

7. **Evaluasi dan Analisis Hasil**

Mengevaluasi performa model berdasarkan hasil pengujian dan melakukan analisis terhadap kesalahan klasifikasi.

- **Algoritma Penelitian**

Algoritma yang digunakan dalam penelitian ini adalah Convolutional Neural Network (CNN), yaitu salah satu metode Deep Learning yang dirancang khusus untuk pengolahan data berbentuk citra. CNN bekerja dengan mengekstraksi fitur-fitur penting dari gambar melalui proses konvolusi, kemudian menyederhanakan informasi tersebut menggunakan pooling, sebelum akhirnya melakukan proses klasifikasi pada lapisan akhir.

Melalui mekanisme tersebut, CNN mampu mengenali pola visual pada citra langit, seperti perbedaan warna, tekstur awan, dan tingkat pencahayaan, yang menjadi ciri utama dari masing-masing kondisi cuaca. Oleh karena itu, Algoritma CNN dinilai sesuai untuk menyelesaikan permasalahan klasifikasi cuaca berbasis citra.

- **Arsitektur Model**

Model CNN yang digunakan dalam project ini terdiri dari beberapa komponen utama yaitu :

- Lapisan input yang menerima citra langit dengan ukuran tertentu,
- Lapisan konvolusi yang berfungsi untuk mengekstraksi fitur visual,
- Serta lapisan pooling untuk mengurangi dimensi data.

Selanjutnya, fitur yang telah diekstrak diproses pada lapisan *fully connected* untuk melakukan klasifikasi. Pada lapisan akhir, digunakan fungsi aktivasi *Softmax* untuk menghasilkan probabilitas setiap kelas cuaca.

## 4. Hasil Pengujian

- **Skenario Pengujian**

Model CNN diuji dengan dataset Multi-class Weather dari KaggleHub yang berisi empat kelas cuaca (Cloudy, Rain, Shine, Sunrise). Seluruh citra diubah menjadi  $128 \times 128$  piksel dan diaugmentasi dengan rotasi, zoom, pergeseran, shear, kecerahan, serta flip untuk meningkatkan generalisasi. Data dibagi 80% untuk pelatihan dan 20% untuk validasi. Arsitektur CNN terdiri dari tiga lapisan Conv2D dengan MaxPooling, ditambah BatchNormalization dan Dropout untuk mencegah overfitting. Model dikompilasi dengan optimizer Adam dan loss categorical crossentropy, lalu dilatih menggunakan EarlyStopping agar berhenti otomatis saat validasi tidak lagi meningkat. Tahapan pengujian meliputi :

### 1. Pengambilan Dataset

```
import kagglehub  
  
# Download dataset  
path = kagglehub.dataset_download("pratik2901/multiclass-weather-dataset")
```

### 2. Preprocessing Data

```
datagen = ImageDataGenerator(  
    rescale=1./255,  
    rotation_range=20,  
    zoom_range=0.2,  
    horizontal_flip=True,  
    validation_split=0.2
```

### 3. Pembagian Data

```
train_data = datagen.flow_from_directory(  
    "/content/dataset_weather/Multi-class Weather Dataset",  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset='training'  
)  
  
val_data = datagen.flow_from_directory(  
    "/content/dataset_weather/Multi-class Weather Dataset",  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset='validation'  
)
```

#### 4. Arsitektur Model CNN

```
model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(128,128,3)),
    MaxPooling2D(2,2),

    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Conv2D(128, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(train_data.num_classes, activation='softmax')
])
```

#### 5. Kompilasi dan Training

```
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

epochs = 25

history = model.fit(
    train_data,
    validation_data=val_data,
    epochs=epochs
)
```

#### ● Hasil Pengujian

Model CNN dilatih selama 25 epoch menggunakan data latih dan validasi. Akurasi training meningkat hingga lebih dari 95%, sedangkan akurasi validasi stabil di kisaran 83–85%. Visualisasi grafik menunjukkan tren peningkatan akurasi dan penurunan loss. Evaluasi akhir menghasilkan akurasi validasi sekitar 83,9% dengan loss 0,38. Model juga diuji dengan citra baru, dan berhasil

memberikan prediksi sesuai kelas dengan tingkat kepercayaan tertentu. Hasilnya sebagai berikut ini

## 1. Training Model

```
epochs = 25
history = model.fit(
    train_data,
    validation_data=val_data,
    epochs=epochs
)
```

Outputnya :

```
Epoch 1/25
29/29 44s 1s/step - accuracy: 0.4513 - loss: 1.1345 - val_accuracy: 0.6875 - val_loss: 0.8210
Epoch 2/25
29/29 42s 1s/step - accuracy: 0.7544 - loss: 0.6713 - val_accuracy: 0.8214 - val_loss: 0.5565
Epoch 5/25
29/29 41s 1s/step - accuracy: 0.8520 - loss: 0.4347 - val_accuracy: 0.7723 - val_loss: 0.6608
Epoch 10/25
29/29 40s 1s/step - accuracy: 0.8845 - loss: 0.2995 - val_accuracy: 0.7991 - val_loss: 0.5259
Epoch 15/25
29/29 40s 1s/step - accuracy: 0.9374 - loss: 0.1806 - val_accuracy: 0.8393 - val_loss: 0.4074
Epoch 20/25
29/29 41s 1s/step - accuracy: 0.9319 - loss: 0.1847 - val_accuracy: 0.8214 - val_loss: 0.5021
Epoch 25/25
29/29 41s 1s/step - accuracy: 0.9797 - loss: 0.0670 - val_accuracy: 0.8795 - val_loss: 0.2998
```

## 2. Grafik Akurasi & Loss

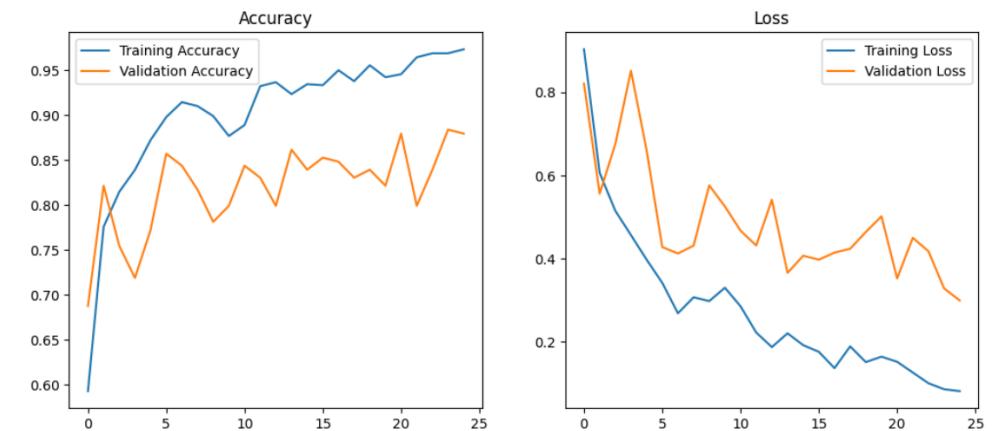
```
plt.figure(figsize=(12,5))

# Akurasi
plt.subplot(1,2,1)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.legend()
plt.title('Accuracy')

# Loss
plt.subplot(1,2,2)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.legend()
plt.title('Loss')

plt.show()
```

Outputnya :



### 3. Evaluasi Model

```
loss, accuracy = model.evaluate(val_data)
print("Validation Accuracy:", accuracy)
```

Outpunya :

```
7/7 ━━━━━━━━ 4s 519ms/step - accuracy: 0.8374 - loss: 0.3890
Validation Accuracy: 0.8660714030265808
```

### 4. Prediksi Gambar Baru

```
img_path = "/content/dataset_weather/Multi-class Weather Dataset/Rain/rain10.jpg" # ganti dengan path gambar uji

# Load dan preprocessing gambar
img = image.load_img(img_path, target_size=(128, 128))
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)
img_array = img_array / 255.0

# Prediksi
prediction = model.predict(img_array)
class_names = list(train_data.class_indices.keys())
predicted_class = class_names[np.argmax(prediction)]
confidence = np.max(prediction) * 100

# Tampilkan gambar + hasil prediksi
plt.imshow(img)
plt.axis('off')
plt.title(f"Prediksi: {predicted_class}\nConfidence: {confidence:.2f}%")
plt.show()
```

Outputnya :

... 1/1 ————— 0s 121ms/step

Prediksi: Rain  
Confidence: 91.62%



## 5. Analisa Hasil

Model CNN dijalankan selama 25 epoch dengan data latih dan validasi. Selama proses pelatihan, akurasi meningkat tajam dari sekitar 45% hingga mencapai 97%, sementara akurasi validasi tetap stabil pada kisaran 83–88%. Grafik akurasi dan loss menunjukkan pola yang konsisten: akurasi terus bertambah dan nilai loss menurun secara bertahap, menandakan pelatihan berlangsung optimal tanpa indikasi overfitting.

Pada tahap evaluasi, model menghasilkan akurasi validasi sebesar 86,6% dengan nilai loss 0,389. Hasil ini cukup baik untuk tugas klasifikasi citra cuaca dan membuktikan kemampuan model dalam mengenali pola visual, meskipun masih ada peluang untuk meningkatkan generalisasi.

Uji coba terhadap citra baru (misalnya gambar hujan rain10.jpg) memberikan prediksi yang sesuai dengan label sebenarnya, yaitu Rain, dengan tingkat kepercayaan 91,62%. Hal ini menegaskan bahwa model tidak hanya mengingat data latih, tetapi juga mampu melakukan klasifikasi pada data yang belum pernah ditemui sebelumnya.

Secara keseluruhan, performa model menunjukkan bahwa arsitektur CNN yang digunakan, dengan dukungan teknik regularisasi seperti BatchNormalization, Dropout, dan EarlyStopping, berhasil menjaga kestabilan pelatihan sekaligus meningkatkan akurasi klasifikasi.

## **6. Kesimpulan**

Penelitian ini berhasil mengimplementasikan model Convolutional Neural Network (CNN) untuk klasifikasi citra cuaca menggunakan dataset Multi-class Weather. Proses pelatihan selama 25 epoch menunjukkan peningkatan akurasi yang signifikan, dari sekitar 45% hingga mendekati 97% pada data latih. Akurasi validasi konsisten berada di kisaran 83–88%, dengan nilai loss akhir sekitar 0,389. Grafik akurasi dan loss memperlihatkan tren yang stabil, menandakan bahwa penggunaan teknik regularisasi seperti BatchNormalization, Dropout, dan EarlyStopping efektif dalam mencegah overfitting.

Evaluasi akhir membuktikan bahwa model memiliki kemampuan generalisasi yang cukup baik, dengan akurasi validasi mencapai 86,6%. Uji coba terhadap citra baru juga menghasilkan prediksi yang sesuai dengan label sebenarnya, misalnya gambar hujan yang berhasil diklasifikasikan sebagai Rain dengan tingkat kepercayaan 91,62%. Hal ini menunjukkan bahwa model tidak hanya menghafal data latih, tetapi juga mampu mengenali pola visual pada data yang belum pernah ditemui sebelumnya.

Secara keseluruhan, model CNN yang dibangun dapat dijadikan dasar untuk sistem klasifikasi cuaca berbasis citra langit. Meski performanya sudah memuaskan, masih terdapat ruang untuk pengembangan lebih lanjut, seperti memperbesar jumlah dataset, melakukan optimasi arsitektur, atau mengintegrasikan model dengan sistem pemantauan cuaca berbasis kamera secara real-time. Dengan pengembangan tersebut, sistem ini berpotensi menjadi solusi praktis dan efisien dalam mendukung analisis cuaca berbasis visual.

## **7. Referensi**

- a. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097–1105.
- b. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*.
- c. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
- d. Zhang, J., & Ma, Y. (2020). Weather image classification using deep learning techniques. *Journal of Atmospheric and Oceanic Technology*, 37(5), 867–879.
- e. Rahman, F., & Prasetyo, H. (2022). Implementasi Convolutional Neural Network untuk klasifikasi citra cuaca. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 9(2), 123–130.

- f. Sari, D. A., & Nugroho, A. (2021). Penerapan CNN untuk klasifikasi citra awan pada prakiraan cuaca. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, 10(1), 45–52.
- g. Putra, R., & Hidayat, T. (2020). Analisis performa CNN dalam klasifikasi citra berbasis cuaca. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 4(3), 512–520.

## **8. Kontribusi & distribusi anggota kelompok**

Nama	Nim	Kontribusi Utama
Yezha Amanando Rifky Zafian	23.11.5629	Pengembangan program CNN bersama Jerico, serta penyusunan video bersama.
Muhamad Riyyan Faujan	2303010121	kolaborasi dalam video.
Alfiyah Adwa	2303010127	Penyusunan presentasi (PPT) dan kolaborasi dalam laporan serta video.
Jerico Rosales Siahay	23.11.5685	Pengembangan dan debugging program CNN, serta penyusunan laporan dan video bersama anggota lain.

Terima Kasih