

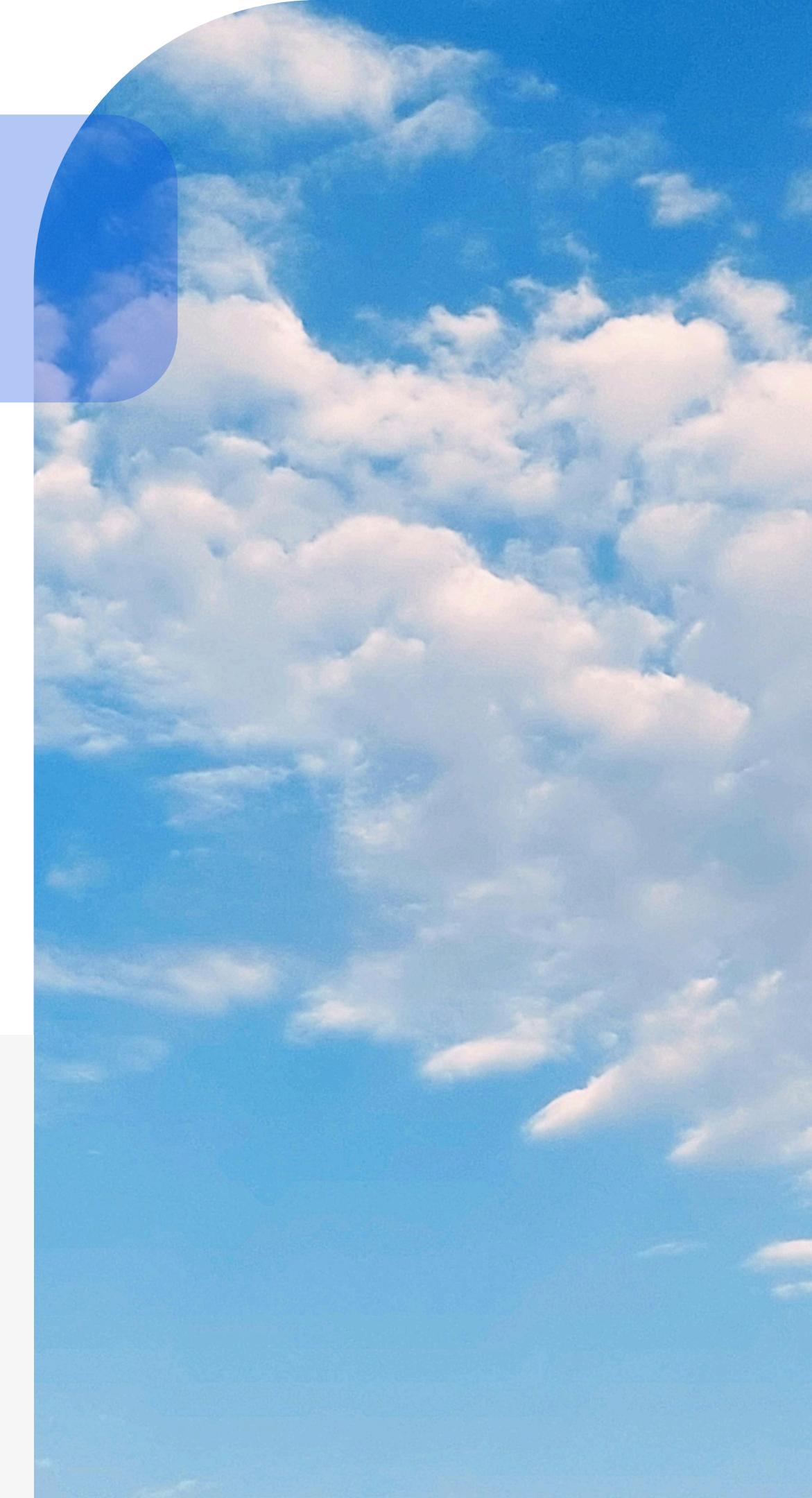


# **FINAL PROJECT KECERDASAN BUATAN (LANJUT)**

**Deteksi Jenis Cuaca dari Citra Langit  
Menggunakan CNN**

**Kelompok 4 :**

1. Yezha amanando rifky zafian(23.11.5629)
2. Muhamad Riyad Faujan(2303010121)
3. Alfiyah Adwa (2303010127)
4. Jerico Rosales Siahay (23.11.5685)





# LATAR BELAKANG



Cuaca merupakan elemen penting yang memengaruhi berbagai aktivitas manusia, seperti transportasi, pertanian, pariwisata, hingga keselamatan. Selama ini, pemantauan cuaca umumnya dilakukan menggunakan perangkat sensor meteorologi yang mengukur suhu, kelembapan, tekanan udara, dan curah hujan. Namun, metode ini memiliki keterbatasan karena membutuhkan peralatan khusus dan tidak tersedia secara merata di semua wilayah. Di sisi lain, pengamatan visual terhadap kondisi langit secara manual juga kurang efektif karena bersifat subjektif dan sulit diterapkan dalam skala besar. Seiring berkembangnya teknologi, khususnya di bidang kecerdasan buatan, metode Convolutional Neural Network (CNN) dapat dimanfaatkan untuk mengenali pola visual pada citra langit secara otomatis dan akurat. Dengan pendekatan ini, sistem klasifikasi cuaca berbasis citra dapat dikembangkan sebagai solusi yang lebih praktis, efisien, dan dapat diandalkan.



# PERMASALAHAN & TUJUAN



01

## PERMASALAHAN:

Permasalahan utama dalam pemantauan cuaca saat ini adalah ketergantungan pada perangkat sensor meteorologi yang membutuhkan biaya tinggi dan belum tersedia secara merata di seluruh wilayah. Selain itu, metode pengamatan visual secara manual terhadap kondisi langit masih bersifat subjektif dan kurang efisien, terutama ketika harus menangani data dalam jumlah besar. Hal ini menyulitkan proses pemantauan cuaca yang cepat, akurat, dan dapat diandalkan dalam berbagai situasi.

02

## TUJUAN:

- Membangun model Convolutional Neural Network (CNN) untuk mendekripsi jenis cuaca dari citra langit
- Mengklasifikasikan cuaca ke dalam beberapa kategori seperti cerah, berawan, hujan, dan matahari terbit
- Mengevaluasi performa model menggunakan data validasi
- Menampilkan hasil prediksi cuaca dari citra baru

# PROGRAM: PENGAMBILAN DATASET

T



Sebelum membangun model CNN, langkah pertama yang kami lakukan adalah mengambil dataset yang akan digunakan sebagai data latih dan validasi.. Dalam proyek ini, kami menggunakan dataset citra langit dari Kaggle, yang berisi gambar dengan empat kategori cuaca: Cloudy, Rain, Shine, dan Sunrise.

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

(untuk outputnya nanti ditunjukan dan dijelaskan di program)

Fungsi dari kode ini adalah Mengunduh dataset citra langit dari Kaggle secara langsung ke dalam proyek. Dataset ini berisi gambar cuaca dengan 4 kelas: Cloudy, Rain, Shine, dan Sunrise.

# PREPROCESSING DATA



Sebelum digunakan untuk pelatihan model, dataset melalui beberapa tahap pra-pemrosesan, yaitu:

- Penyesuaian ukuran citra agar seragam
- Normalisasi nilai piksel untuk mempercepat proses pelatihan
- Pembagian dataset menjadi data latih dan data validasi

Tahapan ini bertujuan agar data lebih siap diproses oleh model CNN dan mengurangi kesalahan saat pelatihan.

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

(untuk outputnya nanti ditunjukan dan dijelaskan di program)

# PEMBAGIAN DATA



Pembagian ini penting agar model tidak hanya belajar dari data yang sama, tetapi juga diuji pada data yang belum pernah dilihat sebelumnya.

Kami menggunakan fungsi `flow_from_directory()` dari `ImageDataGenerator` untuk membagi data secara otomatis, dengan rasio 80% untuk training dan 20% untuk validasi. Selain itu, ukuran gambar disesuaikan menjadi 128x128 piksel agar konsisten dengan input model.

```
train_data = datagen.flow_from_directory(... subset='training')
val_data = datagen.flow_from_directory(... subset='validation')
```

(untuk outputnya nanti ditunjukan dan dijelaskan di program)

# ARSITEKTUR MODEL CNN

T

Model yang digunakan adalah Convolutional Neural Network (CNN) yang dirancang untuk mengenali pola visual dari citra langit. Arsitektur model terdiri dari beberapa lapisan utama yang disusun secara bertahap untuk mengekstraksi fitur dan melakukan klasifikasi ke dalam empat kelas cuaca.

Model ini menggunakan tiga blok Conv2D dan MaxPooling2D untuk mengekstraksi fitur visual, diikuti oleh Flatten dan Dense layer untuk proses klasifikasi. Selain itu, ditambahkan Dropout untuk mencegah overfitting.

```
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')
])
```

(untuk outputnya nanti ditunjukan dan dijelaskan di program)



# COMPILE MODEL

T

Sebelum model dilatih, langkah penting yang harus dilakukan adalah meng-compile model. Proses ini menentukan bagaimana model akan belajar dari data, termasuk algoritma optimasi, fungsi loss, dan metrik evaluasi yang digunakan untuk menilai performa model selama pelatihan.

Kami menggunakan optimizer Adam karena efisien dan adaptif, serta loss function categorical crossentropy karena masalah ini merupakan klasifikasi multi-kelas. Metrik yang digunakan adalah akurasi, untuk melihat seberapa tepat model dalam memprediksi kelas gambar.

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

(untuk outputnya nanti ditunjukan dan dijelaskan di program)



# TRAINING MODEL

T

Setelah model selesai dikompilasi, langkah selanjutnya adalah melatih model menggunakan data training dan memvalidasinya dengan data validasi. Proses ini dilakukan selama 25 epoch, di mana model akan belajar mengenali pola dari gambar dan memperbaiki bobotnya berdasarkan kesalahan prediksi.

```
model.compile(  
    optimizer='adam',  
    loss='categorical_crossentropy',  
    metrics=['accuracy'])  
  
history = model.fit(  
    train_data,  
    validation_data=val_data,  
    epochs=25)
```

(untuk outputnya nanti ditunjukan dan dijelaskan di program)

# VISUALISASI HASIL TRAINING

T



Setelah proses pelatihan selesai, performa model divisualisasikan dalam bentuk grafik akurasi dan loss untuk data training dan validasi. Grafik ini membantu mengevaluasi apakah model belajar dengan baik dan apakah terjadi overfitting.

```
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()
```

(untuk outputnya nanti ditunjukan dan dijelaskan di program)

# EVALUASI MODEL



Evaluasi model dilakukan menggunakan data validasi untuk mengukur kemampuan model dalam mengklasifikasikan jenis cuaca yang belum pernah dilihat sebelumnya. Hasil evaluasi menunjukkan bahwa model CNN memiliki akurasi yang cukup baik dan mampu melakukan generalisasi dengan baik, yang diperkuat oleh grafik akurasi, grafik loss, serta confusion matrix yang menunjukkan sebagian besar prediksi berada pada kelas yang benar.

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

(untuk outputnya nanti ditunjukan dan dijelaskan di program)



# PREDIKSI GAMBAR



Pada tahap prediksi gambar, model CNN digunakan untuk mengklasifikasikan citra langit baru yang belum pernah digunakan pada proses pelatihan. Gambar input diproses oleh model dan menghasilkan prediksi jenis cuaca beserta tingkat keyakinannya, sehingga dapat diketahui apakah citra tersebut termasuk cuaca cerah, berawan, hujan, atau matahari terbit.

```
from tensorflow.keras.preprocessing import image
import numpy as np
import matplotlib.pyplot as plt

# Path gambar uji
img_path = "/content/dataset_weather/Multi-class Weather Dataset/Rain/rain10.jpg"

# 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 # Normalisasi

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

(untuk outputnya nanti ditunjukan dan dijelaskan di program)

# KESIMPULAN

T

Berdasarkan hasil pengujian, model CNN yang dikembangkan mampu mengklasifikasikan jenis cuaca dari citra langit ke dalam empat kategori dengan hasil yang memuaskan. Nilai akurasi yang tinggi serta penurunan loss selama pelatihan menunjukkan bahwa model berhasil mempelajari pola visual dengan baik dan mampu melakukan generalisasi terhadap data baru. Meskipun masih terdapat beberapa kesalahan prediksi, terutama pada kelas cuaca yang memiliki kemiripan visual, secara keseluruhan model menunjukkan performa yang stabil dan konsisten.

Sistem ini memiliki potensi besar untuk dikembangkan lebih lanjut sebagai dasar sistem pemantauan cuaca berbasis visual di masa depan. Peluang peningkatan akurasi masih terbuka melalui penggunaan dataset yang lebih besar, teknik augmentasi data, atau optimasi arsitektur model secara lebih mendalam.



# TERIMAKASIH

MOHON MAAF APABILA BANYAK  
KESALAHAN DAN KEKURANGAN