

# Final Project Presentation

*Implementasi Deep Learning Menggunakan  
Arsitektur Convolutional Neural Network (CNN)  
untuk Klasifikasi Citra Sampah*

## **KELOMPOK 12**

Zaky Naufal 'Alim Budiansyah

Rafi Arsyad

Farel Arga Putra Songgo

Michael Jehezkiel Herjuno W

# TUJUAN PROYEK :

- Membangun Model Deep Learning: Mengimplementasikan arsitektur Convolutional Neural Network (CNN) dari awal (scratch) untuk tugas klasifikasi citra.
- Otomasi Klasifikasi Sampah: Membuat sistem yang mampu membedakan sampah kategori Organic dan Recyclable secara otomatis dengan akurasi tinggi.
- Evaluasi Arsitektur: Mengukur efektivitas model CNN kustom pada dataset publik dengan sumber daya komputasi terbatas.

# Dataset & Proses Training

- Pembagian Data: Dataset dibagi dengan rasio 80% Training (18.052 gambar) dan 20% Validation (4.512 gambar).
- Preprocessing: Citra di-resize menjadi dimensi 150x150 piksel dan dinormalisasi (rescaling  $1/255$ ).
- Data Augmentation: Menerapkan Random Flip, Rotation (0.1), & Zoom (0.1) untuk mencegah overfitting.
- Konfigurasi Model: Menggunakan optimizer Adam dengan Loss Function Sparse Categorical Cross-entropy.

```

img_size = (150, 150)
batch_size = 32

train_ds = tf.keras.utils.image_dataset_from_directory(
    train_dir,
    validation_split=0.2,
    subset="training",
    seed=42,
    image_size=img_size,
    batch_size=batch_size
)

val_ds = tf.keras.utils.image_dataset_from_directory(
    train_dir,
    validation_split=0.2,
    subset="validation",
    seed=42,
    image_size=img_size,
    batch_size=batch_size
)

class_names = train_ds.class_names
print("Class names:", class_names)

```

```

Found 22564 files belonging to 2 classes.
Using 18052 files for training.
Found 22564 files belonging to 2 classes.
Using 4512 files for validation.
Class names: ['O', 'R']

```

- Image Resizing (150x150 piksel) Menyamakan dimensi seluruh gambar input agar matriks pikselnya seragam saat masuk ke model.
- Dataset Splitting (80:20) Membagi total 22.564 gambar secara otomatis:
  - 80% Training: 18.052 gambar untuk melatih model.
  - 20% Validation: 4.512 gambar untuk memvalidasi performa saat pelatihan.
- Batch Processing (Size: 32) Data dimuat dan diproses secara bertahap (per 32 gambar) untuk menjaga efisiensi memori (RAM).
- Auto-Labeling Label kelas (Organic & Recyclable) dideteksi secara otomatis berdasarkan nama folder penyimpanan.

```
plt.figure(figsize=(8,8))
for images, labels in train_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3,3,i+1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")
plt.show()
```



Visualisasi ini menampilkan 9 sampel acak dari dataset pelatihan untuk memverifikasi kesesuaian antara citra dan label kelasnya. Terlihat bahwa dataset memiliki kompleksitas visual yang tinggi dengan variasi pencahayaan, latar belakang, dan posisi objek yang beragam. Sampel mencakup representasi kelas Organik (sayur, buah) dan Recyclable (botol, kertas, plastik). Keragaman visual ini sangat penting untuk melatih model agar mampu melakukan generalisasi dengan baik pada kondisi dunia nyata.



Model: "sequential\_1"

Layer (type)	Output Shape	Param #
sequential (Sequential)	(None, 150, 150, 3)	0
rescaling (Rescaling)	(None, 150, 150, 3)	0
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten (Flatten)	(None, 36992)	0
dense (Dense)	(None, 128)	4,735,104
dense_1 (Dense)	(None, 2)	258

**Total params:** 4,828,610 (18.42 MB)  
**Trainable params:** 4,828,610 (18.42 MB)  
**Non-trainable params:** 0 (0.00 B)

Pada bagian awal, lapisan Rescaling digunakan untuk menormalisasi nilai piksel gambar. Kemudian terdapat tiga blok Conv2D + MaxPooling2D yang berfungsi untuk mengekstrak fitur gambar secara bertahap: mulai dari 32 filter, 64 filter, hingga 128 filter. Setiap lapisan pooling mengurangi dimensi gambar untuk perhitungan yang lebih efisien. Setelah fitur diekstraksi, lapisan Flatten mengubah output menjadi vektor dengan 36.992 node, yang kemudian diproses oleh lapisan Dense dengan 128 neuron.

Akhirnya, lapisan output Dense (2 neuron, softmax) menghasilkan prediksi untuk dua kelas. Jumlah total parameter model adalah 4.828.610, dan semuanya dapat dilatih, artinya semua bobot akan diperbarui selama pelatihan.

```
history = model.fit(  
    train_ds,  
    validation_data=val_ds,  
    epochs=3  
)
```

Epoch 1/3

200/200 ————— 1041s 5s/step - accuracy: 0.7092 - loss: 0.7129 - val\_accuracy: 0.7994 - val\_loss: 0.4589

Epoch 2/3

200/200 ————— 403s 2s/step - accuracy: 0.8128 - loss: 0.4342 - val\_accuracy: 0.8256 - val\_loss: 0.4169

Epoch 3/3

200/200 ————— 398s 2s/step - accuracy: 0.8231 - loss: 0.4070 - val\_accuracy: 0.8100 - val\_loss: 0.4315

### Konfigurasi Pelatihan

- Model dilatih menggunakan fungsi model.fit()
- Data yang digunakan terdiri dari data latih (train\_ds) dan data validasi (val\_ds)
- Proses pelatihan dilakukan selama 3 epoch

### Hasil Pelatihan (Epoch 1–3)

- Akurasi pelatihan meningkat dari 70,9% pada epoch pertama menjadi sekitar 82,3% pada epoch ketiga
- Akurasi validasi berada pada kisaran 79%–82%, dengan nilai akhir sekitar 81%
- Nilai loss pada data latih dan validasi mengalami penurunan, menandakan proses pembelajaran berjalan dengan baik
- Perbedaan antara akurasi pelatihan dan validasi relatif kecil

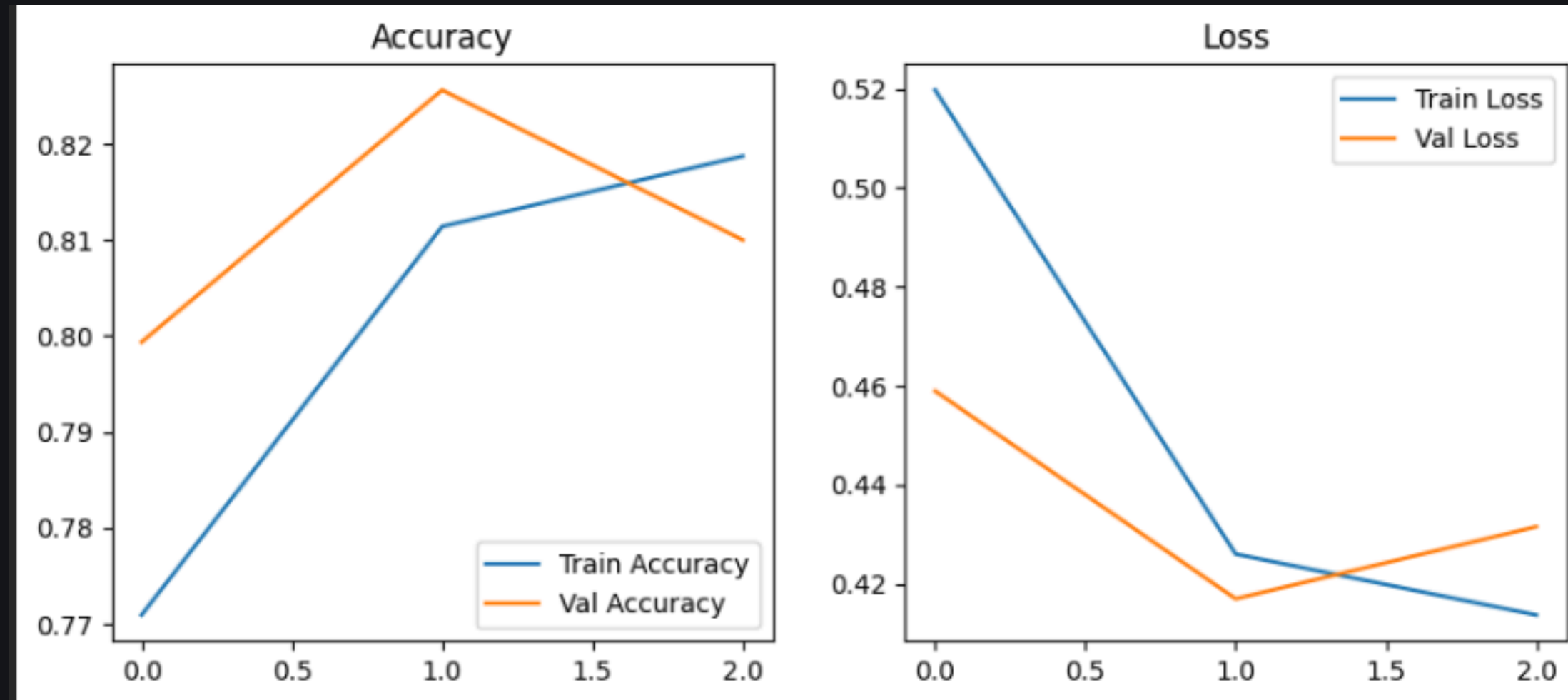
## **Analisis dan Interpretasi**

Hasil pelatihan menunjukkan bahwa model Convolutional Neural Network (CNN) mampu mempelajari pola citra sampah secara efektif. Peningkatan akurasi dan penurunan loss pada setiap epoch menandakan bahwa model semakin baik dalam melakukan klasifikasi. Nilai akurasi pelatihan dan validasi yang seimbang menunjukkan bahwa model tidak mengalami overfitting yang signifikan. Secara keseluruhan, model memiliki kinerja yang stabil dan layak digunakan untuk tugas klasifikasi citra sampah dua kelas pada dataset yang digunakan.

## **Kesimpulan**

Model CNN berhasil dilatih dengan performa yang baik dan stabil, sehingga dapat digunakan sebagai solusi dalam sistem klasifikasi citra sampah.



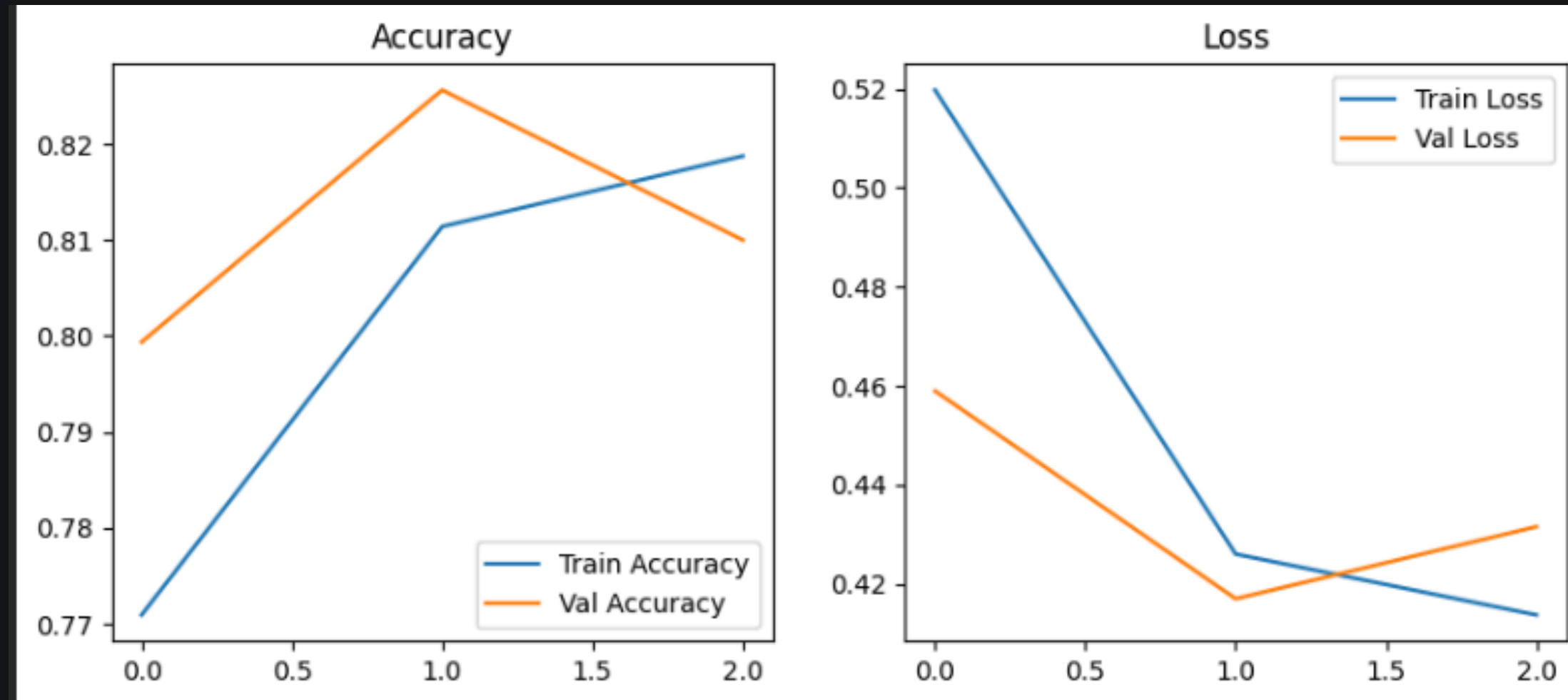


Grafik Akurasi

- Akurasi training meningkat secara konsisten dari epoch 1 hingga epoch 3
- Akurasi validasi meningkat pada epoch 2 dan sedikit menurun pada epoch 3
- Perbedaan antara akurasi training dan validasi relatif kecil
- Hal ini menunjukkan model mampu menggeneralisasi data dengan baik dan tidak mengalami overfitting yang signifikan

Grafik Loss

- Nilai training loss menurun di setiap epoch, menandakan model semakin optimal dalam meminimalkan kesalahan
- Validation loss menurun hingga epoch 2, kemudian sedikit meningkat pada epoch 3
- Kenaikan kecil pada validation loss masih dalam batas wajar dan menunjukkan model mulai mendekati kondisi optimal



### Analisis Grafik Accuracy dan Loss

Grafik menunjukkan bahwa akurasi training meningkat secara konsisten hingga mencapai sekitar 82%, sementara akurasi validasi berada di kisaran 80–82%, menandakan model mampu melakukan generalisasi dengan baik. Nilai training loss menurun di setiap epoch, menunjukkan proses pembelajaran berjalan optimal, sedangkan validation loss sedikit meningkat di akhir epoch namun masih dalam batas wajar. Secara keseluruhan, model CNN memiliki kinerja yang stabil dan tidak mengalami overfitting yang signifikan, sehingga layak digunakan untuk klasifikasi citra sampah.

```
test_loss, test_acc = model.evaluate(test_ds)
print("Test accuracy:", test_acc)
```

```
50/50 ————— 221s 4s/step - accuracy: 0.8710 - loss: 0.4081
Test accuracy: 0.8643749952316284
```

### Evaluasi Model Menggunakan Data Uji (Test Data).

Model CNN dievaluasi menggunakan data uji (test dataset) yang tidak digunakan pada proses pelatihan maupun validasi. Hasil evaluasi menunjukkan akurasi sebesar  $\pm 86\%$ , dengan nilai loss sekitar 0,41. Nilai ini menandakan bahwa model mampu menggeneralisasi dengan baik pada data baru, sehingga performanya tidak hanya baik pada data latih, tetapi juga pada data yang benar-benar belum pernah dilihat sebelumnya. Hal ini memperkuat kesimpulan bahwa model CNN andal dan layak digunakan untuk klasifikasi citra sampah.

```
plt.figure(figsize=(8,8))
for images, labels in test_ds.take(1):
    predictions = model.predict(images)
    for i in range(9):
        ax = plt.subplot(3,3,i+1)
        plt.imshow(images[i].numpy().astype("uint8"))
        pred_label = class_names[np.argmax(predictions[i])]
        true_label = class_names[labels[i]]
        plt.title(f"Pred: {pred_label} | True: {true_label}")
        plt.axis("off")
plt.show()
```

Kode ini mengambil satu batch data dari dataset uji, lalu memprediksi semua gambar dalam batch tersebut menggunakan model CNN. Terus, sembilan gambar pertama ditampilkan dalam grid 3×3, masing-masing disertai dengan informasi label prediksi dan label asli. Tujuan kode ini adalah untuk mengevaluasi hasil prediksi model secara visual dan melihat apakah model mengklasifikasikan gambar dengan benar.



Pred: R | True: R



Pred: O | True: O



Pred: R | True: R



Pred: O | True: O



Pred: O | True: O



Pred: O | True: R



Pred: O | True: O



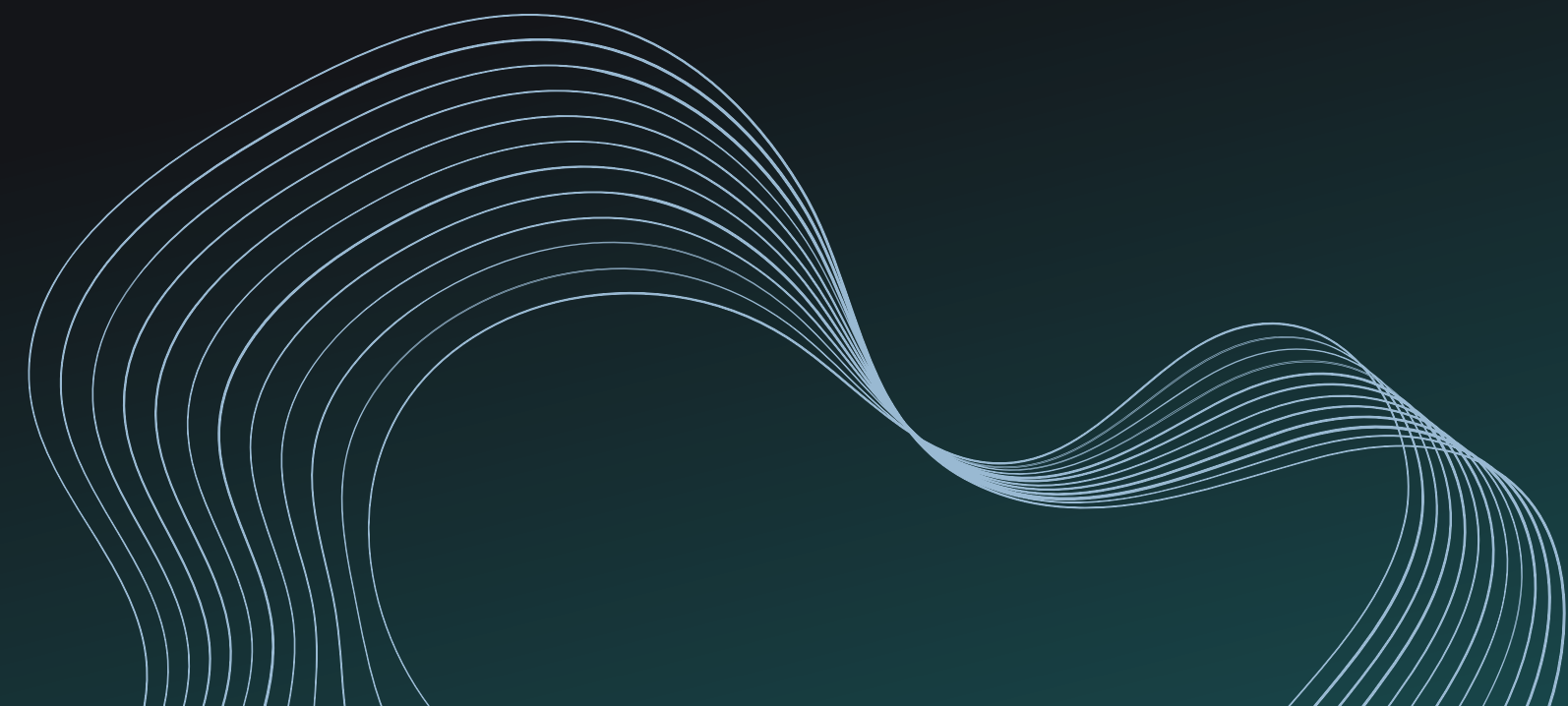
Pred: O | True: O



Pred: O | True: O



Output menampilkan sembilan gambar uji beserta perbandingan antara prediksi model (Pred) dan label sebenarnya (True). Sebagian besar gambar diprediksi dengan benar karena nilai Pred sama dengan True, menunjukkan bahwa model berfungsi dengan baik. Namun, masih ada satu kesalahan klasifikasi, menunjukkan bahwa model belum sempurna.





# Kesimpulan

- Model CNN berhasil membedakan objek kelas O dan R dengan akurasi sekitar 86%.
- Model menunjukkan kemampuan generalisasi yang baik pada dataset uji.
- Hasil prediksi visual menunjukkan sebagian besar gambar diprediksi dengan benar.
- Proyek ini layak digunakan sebagai dasar sistem klasifikasi citra sederhana.



@reallygreatsite

# Thank You

Any Question?