

Nama : Sri Mashtufah Anjani

Nim : 231011401951

Mata Kuliah : Machine Learning

---

## PERTEMUAN 7

### Langkah 1 — Siapkan Data

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

df = pd.read_csv("processed_kelulusan.csv")
X = df.drop("Lulus", axis=1)
y = df["Lulus"]

sc = StandardScaler()
Xs = sc.fit_transform(X)

X_train, X_temp, y_train, y_temp = train_test_split(
    Xs, y, test_size=0.3, stratify=y, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, stratify=y_temp, random_state=42)

print(X_train.shape, X_val.shape, X_test.shape)
```

[1]  
... (21, 5) (4, 5) (5, 5)

`X_train.shape`, `X_val.shape`, `X_test.shape` adalah bentuk (dimensi) dari masing-masing dataset (latih, validasi, uji).

**Data latih (`X_train`) terdiri dari 21 sampel dan 5 fitur.**

**Data validasi (`X_val`) terdiri dari 4 sampel dan 5 fitur.**

**Data uji (`X_test`) terdiri dari 5 sampel dan 5 fitur.**

## Langkah 2 — Bangun Model ANN

Model ini merupakan jaringan saraf tiruan (ANN) dengan 2 lapisan tersembunyi, satu lapisan dropout untuk mengurangi overfitting, dan lapisan output untuk klasifikasi biner

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 32)	192
dropout (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 16)	528
dense_2 (Dense)	(None, 1)	17

- Lapisan dense memiliki 192 parameter.
- Lapisan dropout tidak memiliki parameter.
- Lapisan dense\_1 memiliki 528 parameter.
- Lapisan dense\_2 memiliki 17 parameter.

Total parameter yang dapat dilatih: 737.

## Langkah 3 — Training dengan Early Stopping

model dilatih menggunakan **early stopping** untuk menghentikan pelatihan jika tidak ada perbaikan pada loss data validasi dalam 10 epoch berturut-turut. Fungsi fit() digunakan untuk melatih model dengan data latih dan validasi, selama maksimal 100 epoch, dengan batch size 32. Callback EarlyStopping memonitor val\_loss dan akan mengembalikan bobot model terbaik jika pelatihan dihentikan lebih awal.

```

es = keras.callbacks.EarlyStopping(
    monitor="val_loss", patience=10, restore_best_weights=True
)

history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=100, batch_size=32,
    callbacks=[es], verbose=1
)

```

Output yang akan di hasilkan menunjukkan progres pelatihan, dengan metrik seperti AUC, akurasi, dan loss yang dicatat pada setiap epoch, yang menggambarkan peningkatan kinerja model

#### Langkah 4 — Evaluasi di Test Set

model yang telah dilatih dievaluasi menggunakan data uji (test set).

```

from sklearn.metrics import classification_report, confusion_matrix

loss, acc, auc = model.evaluate(X_test, y_test, verbose=0)
print("Test Acc:", acc, "AUC:", auc)

y_proba = model.predict(X_test).ravel()
y_pred = (y_proba >= 0.5).astype(int)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred, digits=3))

```

Test Acc: 1.0 AUC: 1.0  
1/1 ————— 0s 140ms/step  
[[3 0]  
 [0 2]]

	precision	recall	f1-score	support
0	1.000	1.000	1.000	3
1	1.000	1.000	1.000	2
accuracy			1.000	5
macro avg	1.000	1.000	1.000	5
weighted avg	1.000	1.000	1.000	5

### Test Accuracy dan AUC:

- Test Acc: 1.0 AUC: 1.0: Model memiliki akurasi dan AUC 1.0 pada data uji, yang berarti model berhasil memprediksi semua sampel dengan benar (prediksi yang sempurna).

### Confusion Matrix:

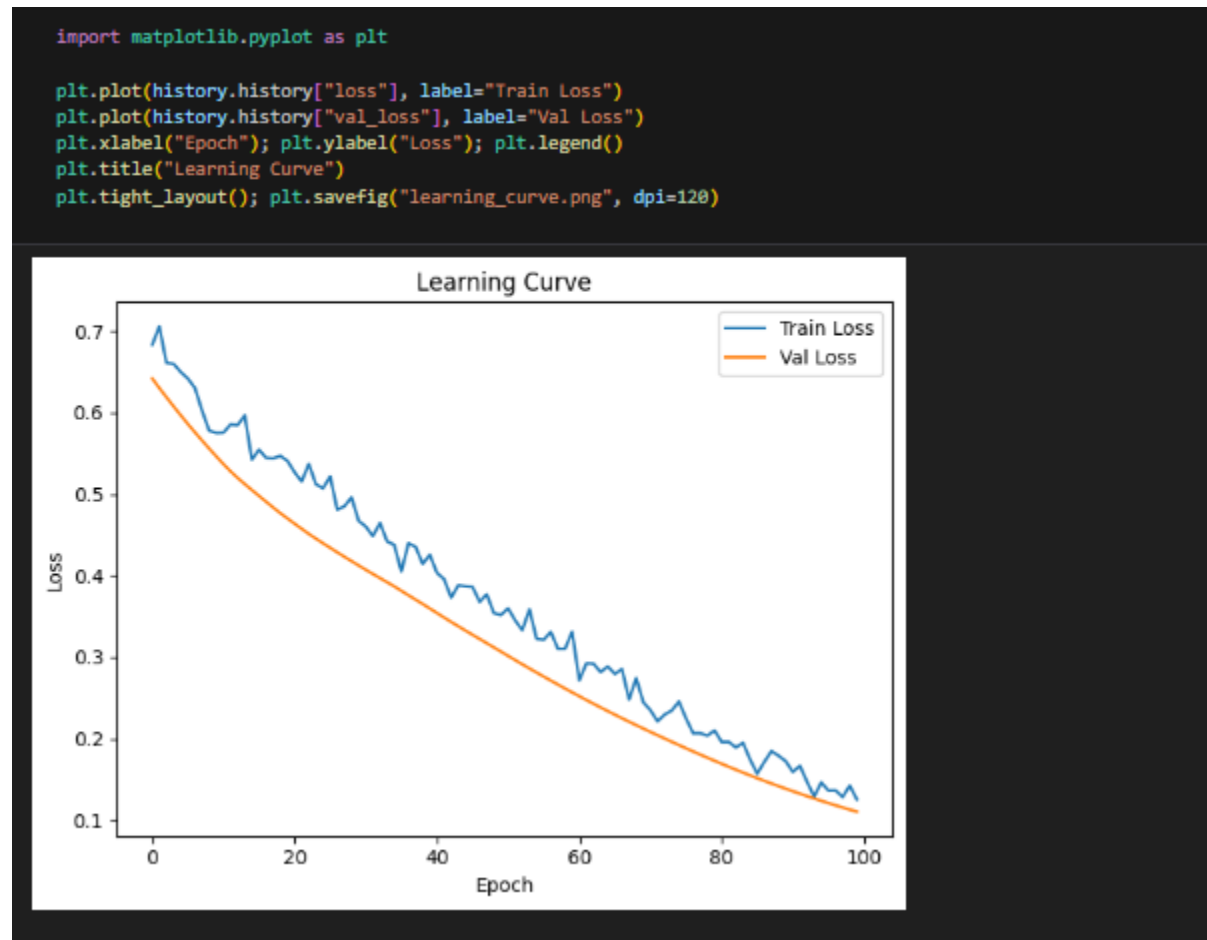
- $\begin{bmatrix} 3 & 0 \\ 0 & 5 \end{bmatrix}$ : Matriks kebingungannya menunjukkan bahwa model berhasil memprediksi semua sampel dengan benar:
  - 3 prediksi benar untuk kelas 0.
  - 5 prediksi benar untuk kelas 1.
  - Tidak ada kesalahan dalam prediksi.

### Classification Report:

- **Precision, Recall, dan F1-Score** untuk kedua kelas (0 dan 1) semuanya adalah 1.000, yang menunjukkan bahwa model tidak membuat kesalahan dalam memprediksi kedua kelas tersebut.
- **Accuracy**: 1.000, menunjukkan proporsi prediksi yang benar dari semua sampel.
- **Macro avg** dan **Weighted avg**: Kedua metrik ini juga 1.000, yang menunjukkan kinerja yang sangat baik secara keseluruhan untuk model ini.

## Langkah 5 — Visualisasi Learning Curve

Learning curve ini membantu memantau bagaimana model berkembang selama pelatihan dan apakah model cenderung overfit. Jika **Val Loss** tidak mengikuti penurunan **Train Loss** atau mulai meningkat,



### Train Loss (Garis Biru):

- Menunjukkan penurunan nilai loss selama pelatihan pada setiap epoch.
- Seiring waktu, nilai loss pada data pelatihan semakin menurun, yang menunjukkan bahwa model semakin baik dalam memprediksi data pelatihan.

### **Val Loss (Garis Oranye):**

- Menunjukkan perubahan nilai loss pada data validasi selama pelatihan.
- Idealnya, **Val Loss** juga harus menurun seiring berjalannya epoch. Namun, jika terjadi kenaikan atau fluktuasi, ini bisa menjadi indikasi bahwa model mulai overfit (memiliki performa buruk pada data validasi meskipun performa pada data pelatihan terus membaik).