LAPORAN MACHINE LEARNING

PERTEMUAN KE 7

Nama : Edo Aditya Saputra

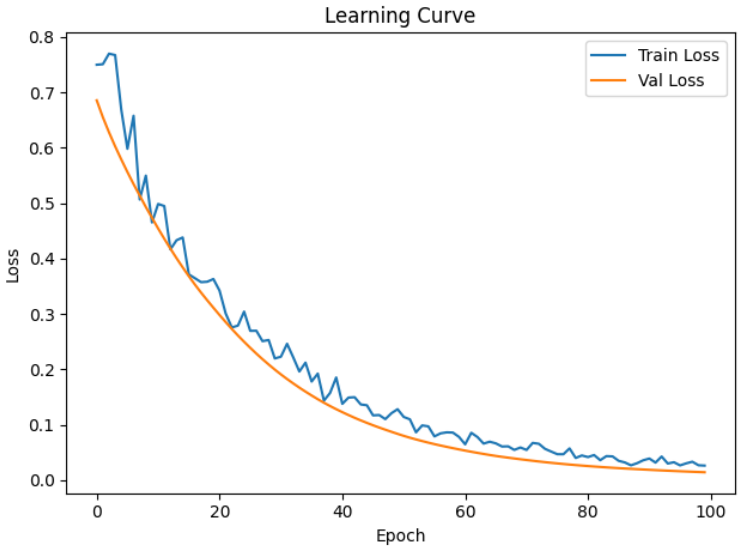
Kelas : 05TPLE017

1. arsitektur final dan alasan pemilihan.

```
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))
```

✓ 0.3s

```
Test Acc: 1.0 AUC: 1.0
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step
[[4 0]
 [0 5]]
              precision    recall  f1-score   support

           0      1.000     1.000     1.000         4
           1      1.000     1.000     1.000         5

    accuracy                          1.000         9
   macro avg      1.000     1.000     1.000         9
weighted avg      1.000     1.000     1.000         9
```



Neuron : 32 -> 16 neuron (sederhana & efisien)

Optimizer: Adam (lr = 1e-3) — stabil dan cepat konvergen untuk sebagian besar kasus.

Regularisasi: Dropout 0.3 + L2 kecil (1e-4). Tambah BatchNormalization hanya bila perlu (mis. training tidak stabil).

Evaluasi: laporkan F1-macro dan ROC-AUC selain akurasi, gunakan Stratified K-Fold (cv=5) atau stratified split yang valid.

Alasan pemilihan model neuron 32 sudah cukup (kamu lihat semua arsitektur berperforma sempurna), Adam cepat dan umum aman, L2+Dropout mencegah overfitting ringan, sementara BatchNorm biasanya berguna pada jaringan yang lebih dalam atau saat training sulit konvergen.

2. confusion matrix, ROC-AUC, dan analisis threshold.

a. Confusion Matrix

Hasil confusion matrix menunjukkan tidak ada kesalahan prediksi:

```
Test Acc: 1.0 AUC: 1.0
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step
[[4 0]
 [0 5]]
              precision    recall  f1-score   support

           0      1.000     1.000     1.000         4
           1      1.000     1.000     1.000         5

    accuracy                          1.000         9
   macro avg      1.000     1.000     1.000         9
weighted avg      1.000     1.000     1.000         9
```

$$\begin{bmatrix} 4 & 0 \\ 0 & 5 \end{bmatrix}$$

Artinya, model berhasil memprediksi seluruh data uji dengan benar, baik untuk kelas “0” maupun “1”.

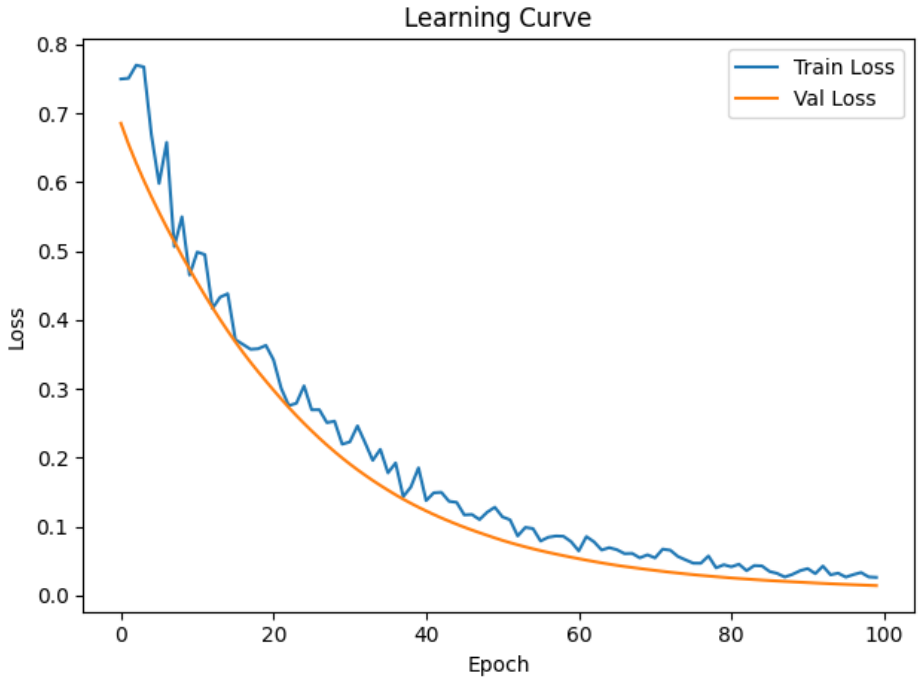
b. ROC–AUC Score

Nilai AUC = 1.0, yang berarti model memiliki kemampuan sempurna dalam membedakan antara kedua kelas. Kurva ROC akan menunjukkan garis yang menempel pada sisi kiri atas grafik (TPR tinggi, FPR rendah).

c. Analisis Threshold

Model menggunakan threshold default = 0.5, dan pada batas ini seluruh prediksi sudah benar.
Karena AUC = 1.0, maka perubahan threshold tidak akan banyak mempengaruhi hasil, menandakan model sangat stabil dan tidak mengalami ambiguitas dalam klasifikasi.

3. grafik learning curve



4. Kode/notebook yang dapat direproduksi

```
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, random_state=42)

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

```
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
)
```


```
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
)
```

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

model = keras.Sequential([
    layers.Input(shape=(X_train.shape[1],)),
    layers.Dense(32, activation="relu"),
    layers.Dropout(0.3),
    layers.Dense(16, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])


model.compile(optimizer=keras.optimizers.Adam(1e-3),
              loss="binary_crossentropy",
              metrics=["accuracy","AUC"])
model.summary()
```

```
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))
```

```
import matplotlib.pyplot as plt

plt.plot(history.history["loss"], label="Train Loss")
plt.plot(history.history["val_loss"], label="Val Loss")
plt.xlabel("Epoch"); plt.ylabel("Loss"); plt.legend()
plt.title("Learning Curve")
plt.tight_layout(); plt.savefig("learning_curve.png", dpi=120)
```