

LAPORAN PERTEMUAN 7

Nama : Muhammad Salman Imamwan Abdillah

Nim : 231011401032

Kelas : 05TPLE017

1. Dataset *processed_kelulusan.csv* dibaca menggunakan pandas dan dipisahkan menjadi fitur (X) serta label target (y). Data fitur kemudian dinormalisasi menggunakan *StandardScaler* agar memiliki skala yang seragam. Selanjutnya, dataset dibagi menjadi tiga bagian yaitu data latih (60%), data validasi (20%), dan data uji (20%) dengan metode *train_test_split* secara stratifikasi untuk menjaga proporsi kelas yang seimbang. Pembagian ini memastikan proses pelatihan dan evaluasi model berlangsung secara adil dan konsisten.

```
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=0.4, 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)
```

✓ 0.1s

.. (6, 5) (2, 5) (2, 5)

2. Model jaringan saraf tiruan dibangun menggunakan TensorFlow Keras dengan arsitektur *Sequential* yang terdiri dari empat lapisan, yaitu dua lapisan *Dense* dengan aktivasi ReLU, satu lapisan *Dropout* untuk mengurangi *overfitting*, dan satu lapisan output dengan aktivasi sigmoid untuk klasifikasi biner. Model dikompilasi menggunakan optimizer Adam dengan *learning rate* 0.001, fungsi *loss* binary crossentropy, serta metrik evaluasi *accuracy* dan *AUC*. Total parameter yang dilatih sebanyak **737**.


```

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
)

```

30.5s

```

1/1 4s 4s/step - AUC: 0.6667 - accuracy: 0.6667 - loss: 0.6543 - val_AUC: 1.0000 - val_accuracy: 1.0000 - val
Epoch 2/100
1/1 0s 164ms/step - AUC: 1.0000 - accuracy: 0.5000 - loss: 0.6333 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 3/100
1/1 0s 154ms/step - AUC: 1.0000 - accuracy: 1.0000 - loss: 0.5298 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 4/100
1/1 0s 161ms/step - AUC: 0.7778 - accuracy: 0.8333 - loss: 0.6532 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 5/100
1/1 0s 162ms/step - AUC: 1.0000 - accuracy: 0.8333 - loss: 0.5576 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 6/100
1/1 0s 159ms/step - AUC: 1.0000 - accuracy: 0.8333 - loss: 0.6215 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 7/100
1/1 0s 169ms/step - AUC: 1.0000 - accuracy: 1.0000 - loss: 0.5416 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 8/100
1/1 0s 320ms/step - AUC: 1.0000 - accuracy: 1.0000 - loss: 0.5292 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 9/100
1/1 0s 247ms/step - AUC: 1.0000 - accuracy: 0.8333 - loss: 0.5522 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 10/100
1/1 0s 295ms/step - AUC: 1.0000 - accuracy: 1.0000 - loss: 0.5365 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 11/100
1/1 0s 193ms/step - AUC: 1.0000 - accuracy: 1.0000 - loss: 0.5569 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 12/100
1/1 0s 257ms/step - AUC: 1.0000 - accuracy: 1.0000 - loss: 0.4675 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 13/100
...
Epoch 99/100
1/1 0s 268ms/step - AUC: 1.0000 - accuracy: 1.0000 - loss: 0.0766 - val_AUC: 1.0000 - val_accuracy: 1.0000 -
Epoch 100/100
1/1 0s 214ms/step - AUC: 1.0000 - accuracy: 1.0000 - loss: 0.0814 - val_AUC: 1.0000 - val_accuracy: 1.0000 -

```

- Evaluasi model dilakukan menggunakan data uji untuk mengukur kinerja akhir dari jaringan saraf tiruan. Hasil evaluasi menunjukkan nilai akurasi dan AUC sebesar **1.0**, yang menandakan model memiliki kemampuan klasifikasi sempurna terhadap data uji. Berdasarkan *confusion matrix*, seluruh data uji berhasil diprediksi dengan benar tanpa kesalahan. Selain itu, *classification report* menunjukkan nilai *precision*, *recall*, dan *f1-score* masing-masing sebesar **1.000** untuk kedua kelas. Meskipun hasil ini menunjukkan performa yang sangat baik, perlu dilakukan pemeriksaan lebih lanjut terhadap ukuran dan keragaman dataset untuk memastikan model tidak mengalami *overfitting* atau bias terhadap data tertentu.

```

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))

```

0.5s

```

... Test Acc: 1.0 AUC: 1.0
1/1 0s 236ms/step
[[1 0]
 [0 1]]

```

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

5. Visualisasi *learning curve* menunjukkan tren penurunan nilai *loss* baik pada data pelatihan maupun data validasi seiring bertambahnya epoch. Penurunan yang konsisten pada kedua kurva menandakan bahwa model mengalami proses pembelajaran yang stabil tanpa tanda-tanda *overfitting*. Pada akhir pelatihan, nilai *loss* mencapai titik minimum, yang menunjukkan bahwa model mampu mempelajari pola data dengan baik dan memiliki performa generalisasi yang optimal terhadap data validasi.

```
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)
```

✓ 1.6s

