

Tugas 1: Tugas 3 Praktikum Mandiri

Noer Muhammad Ayub - 0110222142

¹ Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: noer22142ti@student.nutulfikri.ac.id

1. Menghubungkan Colab dengan Google Drive

```
[14] ✓ 3s
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

from google.colab import drive

Ini artinya kamu mengimpor modul drive yang disediakan oleh Google Colab. Modul ini memungkinkan Colab untuk terhubung dengan akun Google Drive kamu.

drive.mount('/content/drive')

Ini adalah perintah untuk "memasang" (mount) Google Drive ke sistem file Colab. Setelah dijalankan, Colab akan menampilkan link otorisasi dari Google (biasanya muncul prompt).

2. Memanggil data set lewat gdrive dan menampilkannya

```
[17] ✓ 0s
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/praktikum/praktikum3/data/day.csv')
df.head()
```

	int	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985	
2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801	
3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349	
4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562	
5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600	

Next steps: [Generate code with df](#) [New interactive sheet](#)

import pandas as pd

Digunakan untuk mengimpor library pandas, yaitu pustaka Python yang berfungsi untuk membaca, mengolah, dan menganalisis data dalam bentuk tabel (seperti Excel).

```
df = pd.read_csv("/content/drive/MyDrive/praktikum/praktikum3/data/day.csv")
```

Perintah untuk membaca file CSV bernama day.csv yang disimpan di Google Drive pada folder /praktikum/praktikum3/data/. Hasil pembacaan disimpan ke dalam variabel df (dataframe) agar bisa diolah lebih lanjut.

```
df.head()
```

Menampilkan 5 baris pertama dari data df untuk memastikan file berhasil dibaca dan isinya benar.

3. Menentukan variabel independen (X) dan variabel dependen (Y)

```
[18] ✓ Os X = df[['season', 'yr', 'mnth', 'holiday', 'weekday', 'workingday',  
          'weathersit', 'temp', 'atemp', 'hum', 'windspeed']]  
      y = df['cnt']
```

X berisi kolom-kolom yang menjadi faktor atau fitur untuk prediksi

Y berisi kolom target yang akan diprediksi, yaitu jumlah penyewaan sepeda (cnt)

serta memberi konteks bahwa cnt adalah target yang ingin diprediksi dari berbagai fitur lain seperti cuaca, suhu, dan musim.

4. Proses pembuatan dan evaluasi model regresi linear untuk memprediksi jumlah penyewaan sepeda

```
[20] ✓ Os from sklearn.model_selection import train_test_split  
      from sklearn.linear_model import LinearRegression  
      from sklearn.metrics import mean_squared_error, r2_score  
  
      # Split data (80% train, 20% test)  
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)  
  
      # Buat dan latih model  
      model = LinearRegression()  
      model.fit(X_train, y_train)  
  
      # Prediksi  
      y_pred = model.predict(X_test)  
  
      # Evaluasi model  
      print("R² Score:", r2_score(y_test, y_pred))  
      print("MSE:", mean_squared_error(y_test, y_pred))  
  
      R² Score: 0.8276670090367212  
      MSE: 691035.0082022651
```

```
from sklearn.model_selection import train_test_split
```

Baris ini mengimpor fungsi train_test_split dari pustaka scikit-learn (sklearn). Fungsi ini digunakan untuk membagi data menjadi dua bagian:

- data latih (train set) → untuk melatih model,
- data uji (test set) → untuk menguji seberapa baik model bekerja pada data baru.

from sklearn.linear_model import LinearRegression

Mengimpor kelas LinearRegression, yaitu algoritma regresi linear dari *scikit-learn*. Algoritma ini akan digunakan untuk membuat model yang bisa memprediksi nilai y (cnt) berdasarkan variabel x.

from sklearn.metrics import mean_squared_error, r2_score

Mengimpor dua *metrik evaluasi*:

- `mean_squared_error` → mengukur seberapa besar rata-rata kesalahan prediksi model.
- `r2_score` → mengukur seberapa baik model menjelaskan variasi data (semakin mendekati 1 semakin baik).

Split data (80% train, 20% test)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

Data dibagi menjadi:

- 80% untuk pelatihan (train),
- 20% untuk pengujian (test).

`random_state=42` digunakan agar hasil pembagian selalu sama setiap kali dijalankan (agar bisa direproduksi).

model = LinearRegression() | model.fit(X_train, y_train)

Buat dan latih model:

Membuat objek model regresi linear (`model = LinearRegression()`).

Melatih model menggunakan data pelatihan (fit) agar model “belajar” hubungan antara X dan Y.

y_pred = model.predict(X_test)

Prediksi

Setelah model dilatih, baris ini digunakan untuk memprediksi nilai cnt (jumlah penyewaan sepeda) berdasarkan data uji (`X_test`).

print("R² Score:", r2_score(y_test, y_pred)) | print("MSE:", mean_squared_error(y_test, y_pred))

Evaluasi model

Bagian ini menampilkan performa model:

- R^2 Score (Koefisien Determinasi): menunjukkan seberapa besar variasi data yang bisa dijelaskan oleh model.
→ Nilai mendekati 1 berarti model sangat baik.
- MSE (Mean Squared Error): semakin kecil nilainya, semakin akurat modelnya.

R² Score: 0.8276670090367212

MSE: 691035.0082022651

Interpretasi hasil:

- Nilai $R^2 = 0.82$ (82,7%) artinya model bisa menjelaskan sekitar 82% variasi data jumlah penyewaan sepeda (cnt).
- Nilai MSE ≈ 691.035 menunjukkan rata-rata kesalahan prediksi dalam skala jumlah penyewaan.

5. menampilkan hubungan antar variable dataset dalam bentuk heatmap

```
[22] ✓ 1s
import matplotlib.pyplot as plt
import seaborn as sns

# Exclude non-numeric columns before calculating correlation
numeric_df = df.select_dtypes(include=['number'])

plt.figure(figsize=(10,6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title("Korelasi antar variabel")
plt.show()
```

import matplotlib.pyplot as plt

import seaborn as sns

Mengimpor dua library visualisasi:

- matplotlib.pyplot → untuk membuat grafik dasar seperti line chart atau scatter plot.
- seaborn → library visualisasi tingkat tinggi yang membuat grafik lebih cantik dan informatif (dibangun di atas matplotlib).

numeric_df = df.select_dtypes(include=['number'])

Baris ini memilih hanya kolom yang bertipe numerik dari DataFrame df, karena:

- Analisis korelasi hanya bisa dilakukan pada data numerik.
- Kolom non-numerik seperti tanggal atau teks akan otomatis diabaikan.
Contoh: kolom dteday tidak disertakan dalam korelasi.

plt.figure(figsize=(10,6))

Mengatur ukuran grafik agar hasil heatmap tidak terlalu kecil.

Dalam hal ini:

- Lebar grafik = 10 inci
- Tinggi grafik = 6 inci.

sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')

Membuat heatmap korelasi:

- numeric_df.corr() → menghitung nilai korelasi antar setiap kolom numerik. Nilainya berkisar dari -1 sampai 1:
 - 1 = hubungan positif sempurna
 - -1 = hubungan negatif sempurna
 - 0 = tidak ada hubungan.
- annot=True → menampilkan angka nilai korelasi di dalam kotak heatmap.
- cmap='coolwarm' → menentukan warna gradasi dari biru (negatif) ke merah (positif).

plt.title("Korelasi antar variabel")

Memberi judul pada grafik agar pembaca tahu isi visualisasi.

`plt.show()`

Menampilkan grafik heatmap di layar output

