

# Tugas 3.: Tugas Praktikum Mandiri

Oryza Ayunda Putri - 0110224030

Teknik Informatika, STT Terpadu Nurul Fikri, Depok  
E-mail: nasi.tektekmangudin@gmail.com

## 1. Tugas Praktikum Mandiri

### 1.1 Pembacaan Data

Dapat dilihat bahwa terdapat import pandas as pd perintah ini digunakan untuk *mengimport library* pandas ke python, as pd maksudnya adalah pandas diganti pd supaya kalo mau pakai pandas tinggal tulis pd saja.

Terdapat `df = pd.read_csv('./data/data_praktikum/day.csv', sep=',')` perintah ini merupakan hal utama dalam membaca data CSV. `df=` artinya hasil bacaan disimpan pada DataFrame (tabel), `pd.read_csv()` ini merupakan fungsi dari pandas dalam membaca CSV, `'./data/data_praktikum/day.csv'` lokasi file csv dengan posisi naik satu folder dari file notebook lalu menuju ke lokasi folder `data/data_praktikum` dan mengambil file "hour.csv", `sep=','` memberitahu pandas bahwa pemisah di dalam file csv adalah koma, `df.head()` berfungsi untuk menampilkan 5 baris pertama dari DataFrame kalo mau 8 baris pertama berarti `df.head(8)`.

```
import pandas as pd

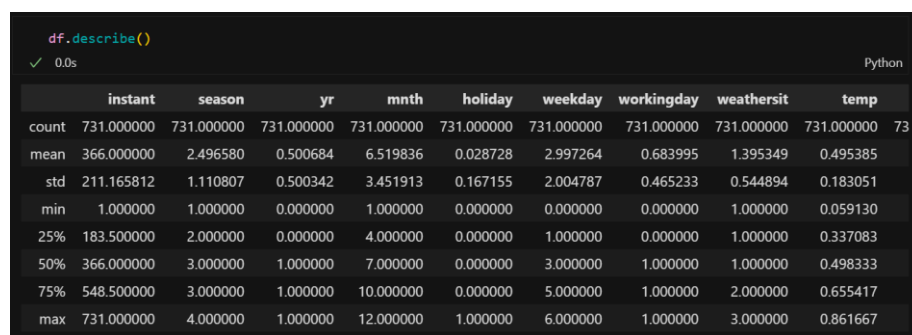
#Membuat data frame untuk membaca data
df = pd.read_csv('../data/day.csv', sep=',')
df.head() # mengambil 5 baris data dari atas
```

0.0sPython

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed
1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446
2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539
3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309
4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296
5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900

Gambar 1 Membaca Dataset

`describe()` menampilkan mean, std, min/max, quartiles.



```
df.describe()
```

	instant	season	yr	mnth	holiday	weekday	workingday	weathersit	temp
count	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000
mean	366.000000	2.496580	0.500684	6.519836	0.028728	2.997264	0.683995	1.395349	0.495385
std	211.165812	1.110807	0.500342	3.451913	0.167155	2.004787	0.465233	0.544894	0.183051
min	1.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.059130
25%	183.500000	2.000000	0.000000	4.000000	0.000000	1.000000	0.000000	1.000000	0.337083
50%	366.000000	3.000000	1.000000	7.000000	0.000000	3.000000	1.000000	1.000000	0.498333
75%	548.500000	3.000000	1.000000	10.000000	0.000000	5.000000	1.000000	2.000000	0.655417
max	731.000000	4.000000	1.000000	12.000000	1.000000	6.000000	1.000000	3.000000	0.861667

Gambar 2 Inspect Data

## 1.2 Data preprocessing

```
# Data Preprocessing untuk dataset Bike Sharing
# Hapus kolom yang tidak relevan untuk prediksi
df1 = df.drop(columns=['instant', 'dteday', 'casual', 'registered']).copy()

# Cek apakah ada nilai kosong
print(df1.isnull().sum())

# Tampilkan 5 baris pertama
df1.head()
```

Gambar 3 Data Preprocessing

Tujuan Penghapusan Data:

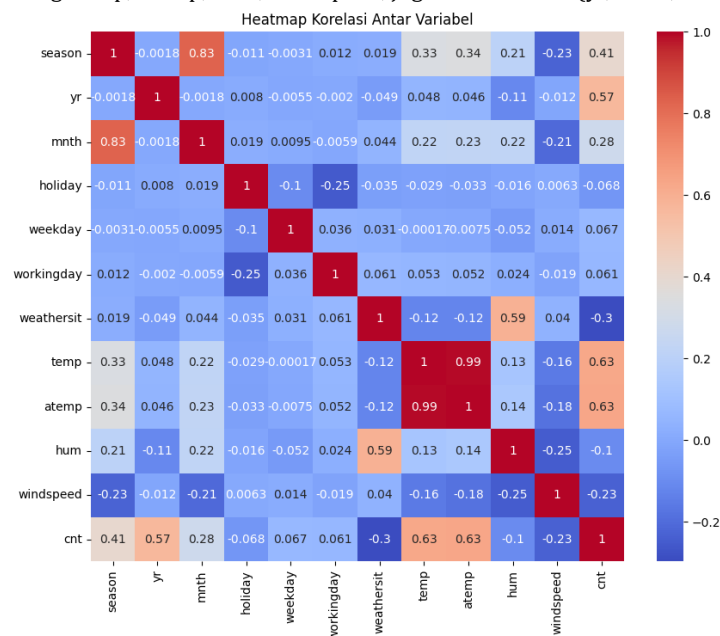
- casual + registered jika dimasukkan → leakage karena cnt = casual + registered.
- dteday bisa diubah menjadi fitur waktu (weekday, month) tetapi dataset sudah menyediakan weekday, mnth, dll.
- Memastikan tidak ada value kosong.

## 1.3 Analisis Korelasi

```
import matplotlib.pyplot as plt
import seaborn as sns
# Membuat heatmap korelasi antar variabel
plt.figure(figsize=(10,8))
sns.heatmap(df1.corr(), annot=True, cmap='coolwarm')
plt.title("Heatmap Korelasi Antar Variabel")
plt.show()
```

Gambar 4 Analisis Korelasi

- Heatmap membantu melihat fitur mana yang berkorelasi kuat dengan cnt.
- Biasanya fitur penting: temp, atemp, hum, windspeed, juga faktor waktu (yr, mnth, season) dan weathersit.



Gambar 5 Output Heatmap

## 1.4 Menentukan X (fitur) dan Y (target)

```
# Variabel Independen (X) dan dependen (y)
X = df[['season', 'yr', 'mnth', 'holiday', 'weekday', 'workingday',
        'weathersit', 'temp', 'atemp', 'hum', 'windspeed']]
y = df['cnt']

y.head()
X.head()
```

	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed
0	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446
1	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539
2	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309
3	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296
4	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900

Gambar 6 Menentukan X (Fitur) dan y (Target)

- X berisi fitur yang dipakai; kamu bisa menambah/kurangi fitur nanti.
- y adalah cnt, target yang diprediksi.

## 1.5 Membagi data (train/test)

```
from sklearn.model_selection import train_test_split

# Membagi 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
)

print("Jumlah data training:", len(X_train))
print("Jumlah data testing :", len(X_test))
```

✓ 0.1s

Jumlah data training: 584  
Jumlah data testing : 147

Gambar 7 Membagi Data

- 80% data untuk melatih, 20% untuk menguji.
- random\_state agar hasil reproducible.

## 1.6 Membuat dan Melatih Model Linier Regression

```
from sklearn.linear_model import LinearRegression

# Inisialisasi model
model = LinearRegression()

# Melatih model
model.fit(X_train, y_train)
```

✓ 0.0s

LinearRegression ⓘ ?

Parameters

Gambar 8 Membuat dan Melatih Model Linier Regression

- fit() mempelajari koefisien dari data training.
- joblib.dump() menyimpan model ke file untuk dipakai lagi.

## 1.7 Prediksi & Evaluasi

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Prediksi data testing
y_pred = model.predict(X_test)

# Menghitung metrik evaluasi
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("R² Score:", r2)
```

Gambar 9 Prediksi dan Evaluasi Model

- **MAE**: rata-rata selisih absolut → mudah diinterpretasikan.
- **MSE**: menekankan outlier (kuadrat error).
- **RMSE**: satuan sama dengan target.
- **R<sup>2</sup>**: proporsi variansi target yang dijelaskan model (0..1, semakin besar lebih baik).

## 1.8 Persamaan regresi (intercept & koefisien)

```
# Menampilkan nilai intercept dan koefisien tiap variabel
print("Intercept:", model.intercept_)

coeff_df = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
coeff_df
```

✓ 0.0s

Intercept: 1248.3289284778209

	Coefficient
season	524.722536
yr	2023.997547
mnth	-38.444658
holiday	-391.550766
weekday	72.937003
workingday	160.804892
weathersit	-632.856284
temp	2097.247836
atemp	3488.042179
hum	-865.439419
windspeed	-2080.540395

Gambar 10 Persamaan Regresi

Dalam regresi linear, model kamu punya bentuk dasar seperti:

$$\text{cnt} = b_0 + b_1X_1 + b_2X_2 + \dots + b_nX_n$$

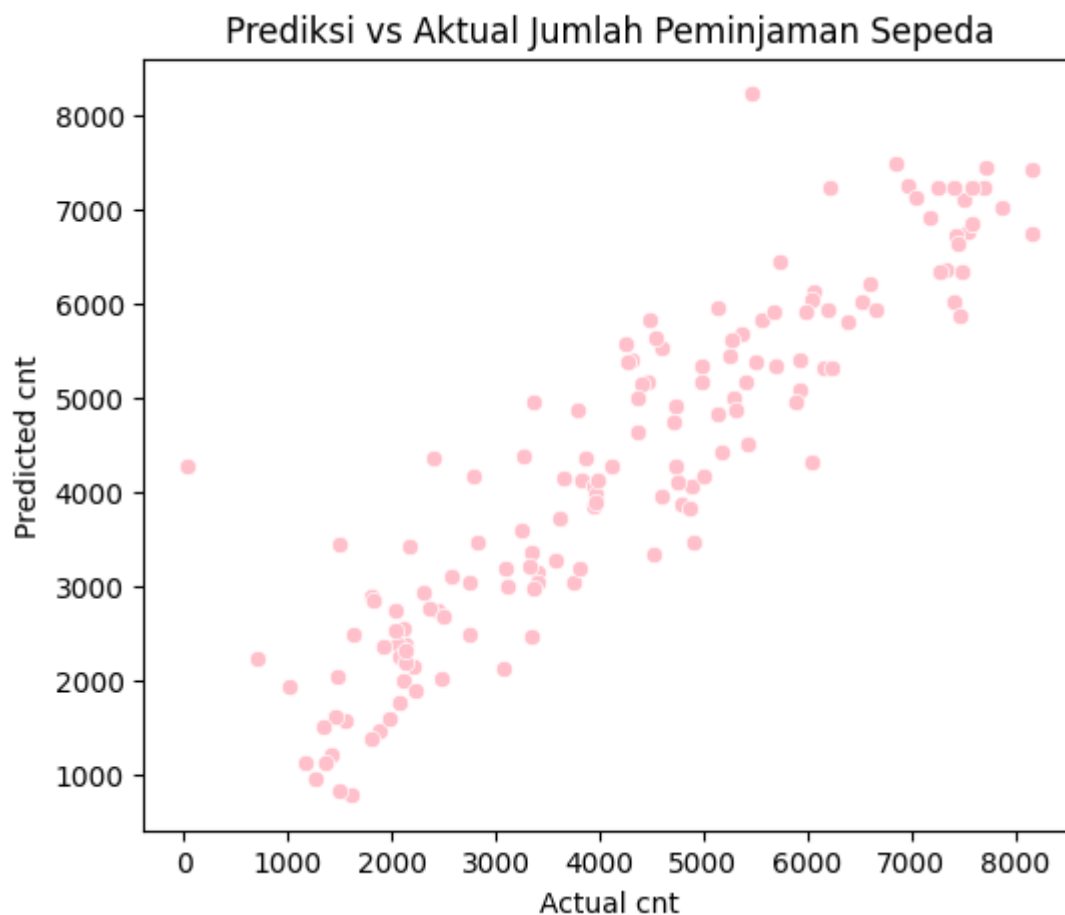
- $b_0$  → **intercept**, atau konstanta model.  
Ini nilai cnt yang diprediksi kalau semua variabel X bernilai nol.
- $b_1, b_2, \dots, b_n$  → **koefisien** untuk tiap variabel (berapa besar pengaruhnya).
- Nilai **positif** → fitur meningkatkan prediksi cnt.  
Contoh: temp = 2431.56 → semakin panas, makin banyak orang bersepeda.
- Nilai **negatif** → fitur menurunkan cnt.  
Contoh: hum = -350.80 → semakin lembap, makin sedikit peminjaman.
- Nilai besar (ribuan) → fitur itu punya pengaruh kuat terhadap hasil prediksi.

## 1.9 Visualisasi: Prediksi vs Aktual

```
plt.figure(figsize=(6,5))
sns.scatterplot(x=y_test, y=y_pred, color='pink')
plt.xlabel("Actual cnt")
plt.ylabel("Predicted cnt")
plt.title("Prediksi vs Aktual Jumlah Peminjaman Sepeda")
plt.show()
```

Gambar 11 Code Visualisasi

- Set ukuran figure.
- `sns.scatterplot(...)` plot titik prediksi vs aktual; `color='pink'` ubah warna titik; `s` ukuran titik; `alpha` transparansi.
- 3–4. Tentukan rentang minimal dan maksimal di kedua sumbu agar garis diagonal pas.
- `plt.plot(...)` menggambar garis diagonal  $y = x$  sebagai referensi ideal (prediksi sempurna akan jatuh di garis ini).
- 6–8. Label dan judul.
- `tight_layout()` agar elemen plot tidak terpotong.
- Menampilkan plot.



Gambar 12 Visualisasi Persebaran

### 1.10 Tabel perbandingan actual vs predicted

```
Perbandingan Actual & Predict

compare_df = pd.DataFrame({'actual': y_test.values, 'predicted': y_pred})
compare_df = compare_df.reset_index(drop=True)
compare_df.head(20)
```

Gambar 13 Tabel Perbandingan Actual vs Predicted

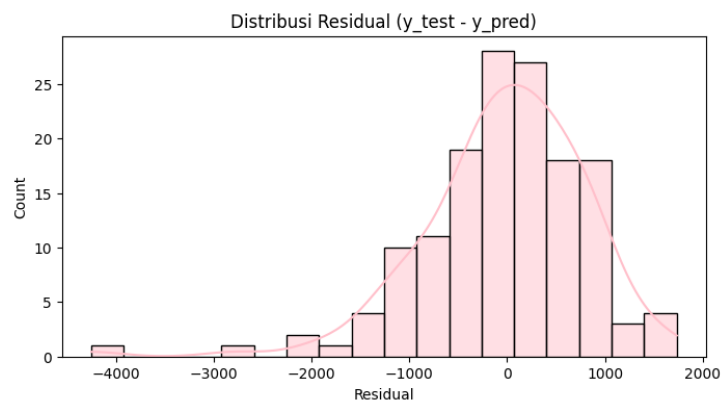
- Buat DataFrame berisi kolom actual dan predicted.
- `reset_index(drop=True)` supaya index baru 0..n.
- Tampilkan 20 baris pertama untuk pemeriksaan manual.

### 1.11 Analisis residual sederhana

```
residuals = y_test.values - y_pred
plt.figure(figsize=(8,4))
sns.histplot(residuals, kde=True, color='pink')
plt.title("Distribusi Residual (y_test - y_pred)")
plt.xlabel("Residual")
plt.show()
```

Gambar 14 Analisis Residual Sederhana

- `residuals = error per sampel`.
  - Set ukuran plot.
  - `histplot` menampilkan histogram dan KDE distribusi residual.
- 4–6. Judul & tampilkan. Jika residual normal dan centered di sekitar 0, tanda baik; jika bias atau heteroskedastisitas, perlu investigasi.



Gambar 15 Analisis Residual Sederhana

### Kesimpulan:

Berdasarkan hasil perhitungan koefisien regresi, diketahui bahwa variabel suhu (`temp`) dan suhu dirasakan (`atemp`) memiliki pengaruh positif terbesar terhadap jumlah peminjaman sepeda (`cnt`). Sebaliknya, kelembapan (`hum`), kecepatan angin

(windspeed), dan kondisi cuaca (weathersit) memberikan pengaruh negatif. Dengan demikian, model regresi linear ini dapat digunakan untuk memperkirakan jumlah peminjaman sepeda berdasarkan kondisi cuaca dan waktu dengan tingkat akurasi yang cukup baik.