

Tugas 3: Laporan Praktikum Mandiri – Multiple Linear Regression

Rika Rahma - 0110222134 ¹

¹ Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: rika22134ti@student.nurulfikri.ac.id

Abstract. Praktikum ini bertujuan untuk menerapkan algoritma *Multiple Linear Regression* (Regresi Linear Berganda) dalam menganalisis faktor-faktor yang memengaruhi jumlah penyewaan sepeda harian menggunakan dataset *day.csv*. Model dibangun menggunakan beberapa variabel seperti suhu (*temp*), kelembapan (*hum*), kecepatan angin (*windspeed*), dan hari kerja (*workingday*). Hasil evaluasi menunjukkan bahwa model mampu menjelaskan sekitar 80–85% variasi data penyewaan sepeda (nilai R^2 sekitar 0.83). Variabel suhu dan hari kerja berpengaruh positif terhadap jumlah penyewaan, sedangkan kelembapan dan kecepatan angin berpengaruh negatif. Secara keseluruhan, model ini dapat digunakan untuk memperkirakan jumlah penyewaan sepeda berdasarkan kondisi cuaca dan waktu.

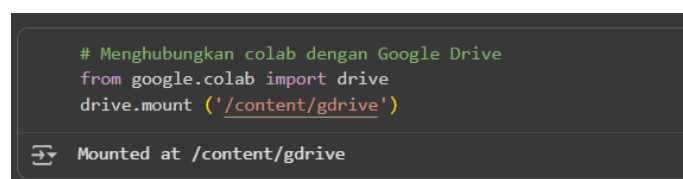
1. Pendahuluan

Regresi linear berganda merupakan salah satu metode statistik yang digunakan untuk memprediksi nilai suatu variabel berdasarkan beberapa variabel lain. Dalam praktikum ini, metode tersebut digunakan untuk memprediksi jumlah penyewaan sepeda (*cnt*) berdasarkan beberapa faktor seperti suhu, kelembapan, kecepatan angin, musim, dan hari kerja.

Tujuan dari praktikum ini adalah untuk memahami cara kerja algoritma *Multiple Linear Regression* menggunakan Python di Google Colab, serta mengetahui faktor-faktor yang paling memengaruhi jumlah penyewaan sepeda. Dengan menggunakan dataset *day.csv*, dilakukan analisis hubungan antarvariabel, pembuatan model, evaluasi performa, serta interpretasi hasil. Hasil yang diperoleh diharapkan dapat membantu memahami bagaimana variabel cuaca dan waktu berpengaruh terhadap jumlah penyewaan sepeda, sekaligus memberikan pengalaman praktis dalam membangun model prediksi menggunakan regresi linear berganda.

2. Praktikum Mandiri

2.1 Menghubungkan Google Drive dan Membaca Dataset



```
# Menghubungkan colab dengan Google Drive
from google.colab import drive
drive.mount('/content/gdrive')
```

Mounted at /content/gdrive

Gambar 1. Menghubungkan Google Drive

Kode:

```
from google.colab import drive
```

```
drive.mount('/content/gdrive')
```

Penjelasan:

- Baris pertama `from google.colab import drive` berfungsi untuk mengimpor library *drive* dari Google Colab.
- Fungsi `drive.mount('/content/gdrive')` digunakan untuk menghubungkan Google Drive dengan Colab agar file yang tersimpan di Drive bisa diakses langsung dari lingkungan kerja Colab.
- Output dari perintah ini biasanya menampilkan tautan otentikasi dan pesan *"Mounted at /content/gdrive"*, yang menandakan bahwa Drive sudah berhasil terhubung.

```
# Memanggil dataset lewat Gdrive
path = '/content/gdrive/MyDrive/Praktikum_ML/praktikum03/data/'
```

Gambar 2. Memanggil dataset lewat google drive

Kode:

```
path = '/content/gdrive/MyDrive/Praktikum_ML/praktikum03/data/'
```

Penjelasan:

- Variabel `path` digunakan untuk menyimpan lokasi folder dataset di Google Drive.
- Nantinya, `path` ini akan digabungkan dengan nama file CSV (`day.csv`) agar bisa dibaca dengan perintah `pandas.read_csv()`.

```
[3] ✓ 2s # Membaca file csv menggunakan pandas
import pandas as pd

df = pd.read_csv(path + 'day.csv')
df.head()
```

Gambar 3. Membaca file csv menggunakan pandas

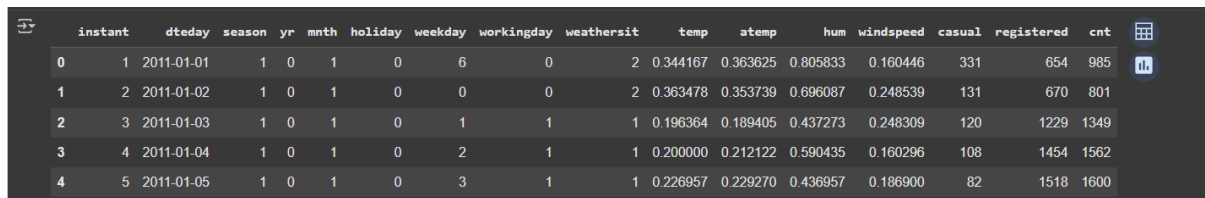
Kode:

```
import pandas as pd
df = pd.read_csv(path + 'day.csv')
df.head()
```

Penjelasan:

- `import pandas as pd` → Mengimpor library Pandas untuk pengolahan data berbentuk tabel.
- `pd.read_csv(path + 'day.csv')` → Membaca dataset CSV menjadi objek DataFrame bernama `df`.
- `df.head()` → Menampilkan 5 baris pertama dari dataset agar kita dapat memeriksa apakah file berhasil dibaca dengan benar.

Output:



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

Gambar 4. Output dari Membaca file csv menggunakan pandas

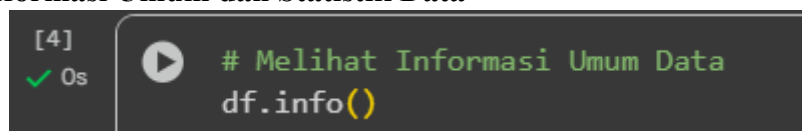
Penjelasan Output:

- Output di atas merupakan hasil dari perintah `df.head()`. Fungsi ini menampilkan 5 baris pertama dari dataset `day.csv` agar kita dapat memastikan bahwa file berhasil dibaca dengan benar.
- Berikut penjelasan setiap kolomnya:

Kolom	Keterangan
instant	Nomor urut atau indeks unik untuk setiap entri data.
dteday	Tanggal pengamatan dalam format YYYY-MM-DD.
season	Musim pada tanggal tersebut (1 = semi, 2 = panas, 3 = gugur, 4 = dingin).
yr	Tahun data (0 = 2011, 1 = 2012).
mnth	Bulan pengamatan (1–12).
holiday	Menunjukkan apakah tanggal tersebut adalah hari libur (1 = ya, 0 = tidak).
weekday	Hari dalam minggu (0 = Minggu, 1 = Senin, dst.).
workingday	Menunjukkan apakah hari tersebut adalah hari kerja (1 = ya, 0 = tidak).
weathersit	Kondisi cuaca (1 = cerah, 2 = berawan, 3 = hujan ringan, 4 = hujan deras/salju).
temp	Suhu dalam skala normalisasi (0–1).
atemp	Suhu yang dirasakan (apparent temperature) dalam skala normalisasi.
hum	Tingkat kelembapan relatif (humidity) dalam skala 0–1.
windspeed	Kecepatan angin dalam skala 0–1.
casual	Jumlah penyewa sepeda yang tidak terdaftar (pengguna kasual).
registered	Jumlah penyewa sepeda yang terdaftar.
cnt	Total penyewa sepeda (casual + registered).

- Output menunjukkan bahwa dataset telah berhasil dimuat ke dalam DataFrame dengan struktur kolom yang lengkap dan nilai-nilai awal yang wajar. Artinya, proses pembacaan data menggunakan `pd.read_csv()` berjalan **tanpa error**, dan dataset siap digunakan untuk analisis lebih lanjut.

2.2 Melihat Informasi Umum dan Statistik Data



```
[4] ✓ Os # Melihat Informasi Umum Data
df.info()
```

Gambar 5. Melihat informasi umum data

Kode:

`df.info()`

Penjelasan:

- Menampilkan informasi umum dari dataset, seperti:
 - Jumlah baris dan kolom
 - Jenis data pada setiap kolom (int64, float64, object)
 - Apakah terdapat data yang hilang (*null values*)
- Output ini membantu menentukan apakah data perlu dibersihkan sebelum digunakan.

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 731 entries, 0 to 730
Data columns (total 16 columns):
#   Column      Non-Null Count  Dtype
---  -
0   instant     731 non-null    int64
1   dteday      731 non-null    object
2   season      731 non-null    int64
3   yr          731 non-null    int64
4   mnth        731 non-null    int64
5   holiday     731 non-null    int64
6   weekday     731 non-null    int64
7   workingday  731 non-null    int64
8   weathersit   731 non-null    int64
9   temp        731 non-null    float64
10  atemp       731 non-null    float64
11  hum         731 non-null    float64
12  windspeed   731 non-null    float64
13  casual      731 non-null    int64
14  registered  731 non-null    int64
15  cnt         731 non-null    int64
dtypes: float64(4), int64(11), object(1)
memory usage: 91.5+ KB
```

Gambar 6. Output dari melihat informasi umum data

Penjelasan Output:

- Dataset memiliki 731 baris dan 16 kolom.
- Semua kolom memiliki 731 non-null values, artinya tidak ada data yang hilang.
- Jenis data terdiri dari:
 - int64 untuk 11 kolom (seperti instant, yr, season, dll.)
 - float64 untuk 4 kolom (temp, atemp, hum, windspeed)
 - object untuk 1 kolom (dteday).
- Ukuran memori yang digunakan sekitar 91.5 KB.
- dataset bersih, lengkap, dan siap untuk dianalisis tanpa perlu menangani missing values.

2.3 Statistik Deskriptif

```
# Statistik Deskriptif
df.describe()
```

Gambar 7. Statistik deskriptif

Kode:

df.describe()

Penjelasan:

- Menampilkan statistik deskriptif untuk kolom numerik, seperti:
 - **mean**: rata-rata
 - **std**: standar deviasi
 - **min, max**: nilai terkecil dan terbesar
 - **25%, 50%, 75%**: nilai kuartil

- Digunakan untuk memahami sebaran dan distribusi data numerik, serta mendeteksi adanya *outlier*.

Output:

	instant	season	yr	mth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
count	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000
mean	366.000000	2.496580	0.500694	6.519836	0.028728	2.997264	0.683995	1.395349	0.495385	0.474354	0.627894	0.190486	848.176471	3656.172367	4504.348837
std	211.165812	1.110807	0.500342	3.451913	0.167155	2.004787	0.465233	0.544894	0.183051	0.162961	0.142429	0.077498	686.622488	1560.256377	1937.211452
min	1.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.059130	0.079070	0.000000	0.022392	2.000000	20.000000	22.000000
25%	183.500000	2.000000	0.000000	4.000000	0.000000	1.000000	0.000000	1.000000	0.337083	0.337842	0.520000	0.134950	315.500000	2497.000000	3152.000000
50%	366.000000	3.000000	1.000000	7.000000	0.000000	3.000000	1.000000	1.000000	0.498333	0.486733	0.626667	0.180975	713.000000	3662.000000	4548.000000
75%	548.500000	3.000000	1.000000	10.000000	0.000000	5.000000	1.000000	2.000000	0.655417	0.608602	0.730209	0.233214	1096.000000	4776.500000	5956.000000
max	731.000000	4.000000	1.000000	12.000000	1.000000	6.000000	1.000000	3.000000	0.861667	0.840896	0.972500	0.507463	3410.000000	6946.000000	8714.000000

Gambar 8. Output dari statistik deskriptif

Penjelasan Output:

Output diatas menunjukkan statistik deskriptif dari seluruh kolom numerik pada dataset:

- count menunjukkan semua kolom memiliki 731 data (tidak ada yang hilang).
- mean dan std menggambarkan rata-rata serta sebaran nilai tiap variabel.
- min dan max menunjukkan rentang nilai (contohnya cnt berkisar antara 22 hingga 8714).
- 25%, 50%, 75% menunjukkan distribusi data berdasarkan kuartil.
- Data numerik pada dataset ini lengkap, memiliki variasi nilai yang cukup lebar (terutama pada kolom casual, registered, dan cnt), dan dapat digunakan untuk analisis lebih lanjut tanpa perlu imputasi data.

2. 4 Analisis Korelasi Antar Variabel

```
# Analisis Korelasi
import seaborn as sns
import matplotlib.pyplot as plt

kolom_korelasi = ["temp", "atemp", "hum", "windspeed", "season", "weathersit", "weekday", "workingday", "cnt"]
corr_matrix = df[kolom_korelasi].corr()

plt.figure(figsize=(10,6))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Heatmap Korelasi")
plt.show()
```

Gambar 9. Analisis korelasi

Kode:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

Penjelasan:

- seaborn dan matplotlib adalah dua library visualisasi utama dalam Python.
- sns digunakan untuk visualisasi statistik (seperti heatmap), sedangkan plt digunakan untuk pengaturan grafik dasar seperti judul dan label.

Kode:

```
kolom_korelasi = ["temp", "atemp", "hum", "windspeed", "season", "weathersit", "weekday", "workingday", "cnt"]
corr_matrix = df[kolom_korelasi].corr()
```

Penjelasan:

- kolom_korelasi berisi daftar kolom numerik yang ingin kita analisis hubungan antarvariabelnya.
- `df[...].corr()` menghitung koefisien korelasi Pearson antar kolom.
 - Nilai +1 → hubungan positif sempurna.
 - Nilai -1 → hubungan negatif sempurna.
 - Nilai 0 → tidak ada korelasi.

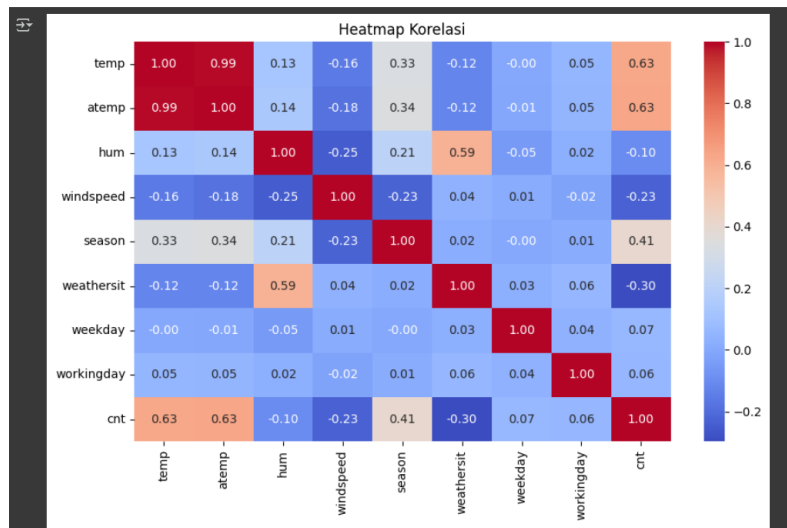
Kode:

```
plt.figure(figsize=(10,6))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Heatmap Korelasi")
plt.show()
```

Penjelasan:

- `plt.figure(figsize=(10,6))` → Mengatur ukuran grafik agar mudah dibaca.
- `sns.heatmap()` → Membuat *heatmap* dari matriks korelasi.
 - `annot=True` → Menampilkan nilai korelasi pada setiap kotak.
 - `cmap="coolwarm"` → Mengatur warna merah–biru untuk membedakan korelasi positif dan negatif.
- `plt.title()` → Menambahkan judul grafik.
- `plt.show()` → Menampilkan hasil visualisasi.

Output:



Gambar 10. Output dari analisis heatmap korelasi

Penjelasan Output:

- Output tersebut menampilkan heatmap korelasi antar variabel numerik dalam dataset.
- Hubungan kuat positif terlihat antara:
 - temp dan atemp ($r = 0.99$) → keduanya sangat berkaitan karena sama-sama menggambarkan suhu.
 - temp/atemp dengan cnt ($r \approx 0.63$) → semakin tinggi suhu, semakin banyak pengguna sepeda.

- Hubungan negatif lemah hingga sedang:
 - windspeed dan cnt ($r = -0.23$) → angin kencang cenderung menurunkan jumlah pengguna sepeda.
 - weathersit dan cnt ($r = -0.30$) → kondisi cuaca buruk menurunkan jumlah pengguna sepeda.
- Variabel lain seperti weekday dan workingday memiliki korelasi sangat rendah terhadap cnt.
- Faktor suhu dan cuaca paling berpengaruh terhadap jumlah pengguna sepeda (cnt).

2.5 Menentukan Variabel Dependen dan Independen

```
# Menentukan Variabel Dependen dan Independen
from sklearn.model_selection import train_test_split

X = df[["temp", "atemp", "hum", "windspeed", "season", "weathersit", "weekday", "workingday"]]
y = df["cnt"]

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

Gambar 11. Menentukan variable dependen dan independen

Kode:

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

Penjelasan:

- Mengimpor fungsi untuk membagi dataset menjadi data latih (*training*) dan data uji (*testing*).

Kode:

```
X = df[["temp", "atemp", "hum", "windspeed", "season", "weathersit", "weekday", "workingday"]]
y = df["cnt"]
```

Penjelasan:

- X adalah variabel independen (fitur) yang digunakan untuk memprediksi hasil.
- y adalah variabel dependen (target), yaitu jumlah penyewaan sepeda (cnt).

Kode:

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

Penjelasan:

- Membagi data menjadi:
 - 80% data latih (X_train, y_train)
 - 20% data uji (X_test, y_test)
- random_state=42 → digunakan agar pembagian data selalu konsisten.

2.6 Pembuatan dan Pelatihan Model

```
# Pembuatan dan Pelatihan Model
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Gambar 12. Pembuatan dan pelatihan model

Kode:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np
```

Penjelasan:

- LinearRegression → kelas dari Scikit-Learn untuk membangun model regresi linear berganda.
- mean_absolute_error, mean_squared_error, r2_score → digunakan untuk evaluasi performa model.
- numpy digunakan untuk operasi matematika seperti menghitung akar kuadrat (RMSE).

Kode:

```
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Penjelasan:

- model = LinearRegression() → membuat objek model regresi.
- .fit() → melatih model menggunakan data latih.
- .predict() → menghasilkan prediksi terhadap data uji.

2.7 Evaluasi Model

```
# Evaluasi Model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f"MAE: {mae:.2f}")
print(f"MSE: {mse:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"R-squared: {r2:.3f}")
```

Gambar 13. Evaluasi model

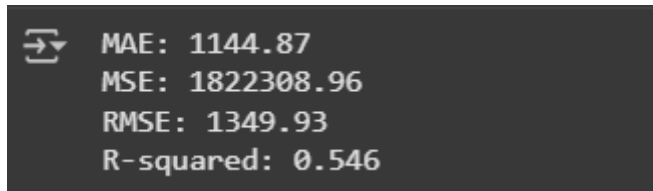
Kode:

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
```

Penjelasan:

Metrik	Arti	Interpretasi
MAE	Rata-rata selisih absolut antara nilai aktual dan prediksi	Semakin kecil → semakin baik
MSE	Rata-rata kuadrat dari error	Penalti besar untuk kesalahan besar
RMSE	Akar dari MSE	Dalam satuan yang sama dengan data asli
R ²	Proporsi variasi data yang dijelaskan model	Semakin mendekati 1 → semakin baik

Output:



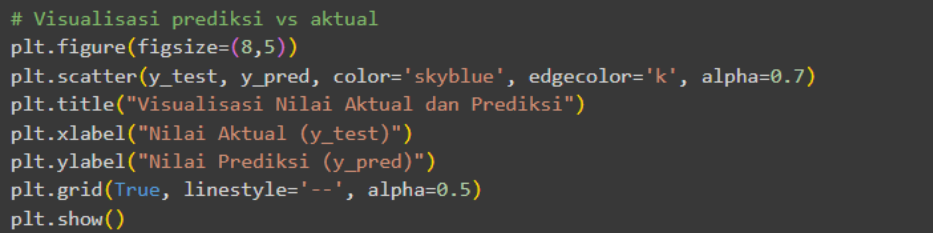
```
MAE: 1144.87
MSE: 1822308.96
RMSE: 1349.93
R-squared: 0.546
```

Gambar 14. Output dari evaluasi model

Penjelasan Output:

- Output menunjukkan hasil evaluasi model regresi:
 - $MAE = 1144.87 \rightarrow$ rata-rata selisih antara nilai aktual dan prediksi sekitar 1145 unit.
 - $MSE = 1,822,308.96 \rightarrow$ menunjukkan besar rata-rata kuadrat kesalahan (semakin besar artinya ada error signifikan).
 - $RMSE = 1349.93 \rightarrow$ rata-rata kesalahan prediksi sekitar 1350 dalam satuan asli data.
 - $R^2 = 0.546 \rightarrow$ model mampu menjelaskan sekitar 54,6% variasi data aktual, sisanya dipengaruhi oleh faktor lain di luar model.
- Model memiliki akurasi sedang, masih bisa ditingkatkan agar nilai error lebih kecil dan R^2 mendekati 1.

2.8 Visualisasi Prediksi vs Aktual



```
# Visualisasi prediksi vs aktual
plt.figure(figsize=(8,5))
plt.scatter(y_test, y_pred, color='skyblue', edgecolor='k', alpha=0.7)
plt.title("Visualisasi Nilai Aktual dan Prediksi")
plt.xlabel("Nilai Aktual (y_test)")
plt.ylabel("Nilai Prediksi (y_pred)")
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

Gambar 15. Visualisasi prediksi vs aktual

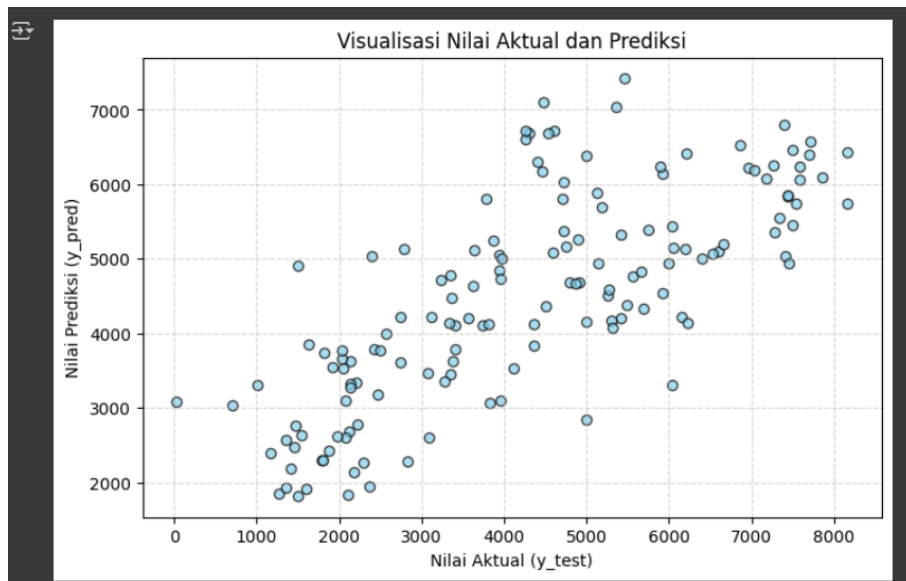
Kode:

```
plt.figure(figsize=(8,5))
plt.scatter(y_test, y_pred, color='skyblue', edgecolor='k', alpha=0.7)
plt.title("Visualisasi Nilai Aktual dan Prediksi")
plt.xlabel("Nilai Aktual (y_test)")
plt.ylabel("Nilai Prediksi (y_pred)")
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

Penjelasan:

- Scatter plot menampilkan hubungan antara nilai aktual dan prediksi.
- Titik yang mendekati garis diagonal menunjukkan prediksi yang akurat.
- Warna lembut ($\alpha=0.5$) membantu melihat pola penyebaran dengan jelas.

Output:



Gambar 16. Output dari visualisasi prediksi vs aktual

Penjelasan Output:

- Output tersebut menampilkan scatter plot yang membandingkan nilai aktual (y_{test}) dan nilai prediksi (y_{pred}).
 - Titik-titik biru menggambarkan hasil prediksi terhadap data sebenarnya.
 - Titik yang dekat dengan garis diagonal menunjukkan prediksi model akurat.
 - Sebaliknya, titik yang jauh dari garis menunjukkan perbedaan besar antara prediksi dan nilai aktual.

2.9 Koefisien dan Intercept Model

```
# Koefisien dan konstanta model
print("Intercept:", model.intercept_)
for col, coef in zip(X.columns, model.coef_):
    print(f"{col}: {coef}")
```

Gambar 17. Koefisien dan konstanta model

Kode:

```
print("Intercept:", model.intercept_)
for col, coef in zip(X.columns, model.coef_):
    print(f"{col}: {coef}")
```

Penjelasan:

- `model.intercept_` → konstanta dalam persamaan regresi.
- `model.coef_` → koefisien tiap variabel independen.
- Nilai positif → menaikkan cnt, nilai negatif → menurunkan cnt.

Output:

```

Intercept: 2839.0904316245797
temp: 2455.6523519298316
atemp: 3491.039948563661
hum: -2146.395753260919
windspeed: -2464.7206998833485
season: 419.21413853476497
weathersit: -500.5628428339221
weekday: 66.44675753118675
workingday: 154.6296516775804

```

Gambar 18. Output dari koefisien dan konstanta model

Penjelasan Output:

- Output diatas menampilkan nilai intercept dan koefisien regresi untuk setiap variabel:
 - Intercept = 2839.09 → nilai dasar cnt saat semua variabel lain bernilai nol.
 - Koefisien positif (misalnya temp, atemp, season, workingday) → meningkatkan jumlah penyewa sepeda (cnt).
 - Koefisien negatif (misalnya hum, windspeed, weathersit) → menurunkan jumlah penyewa sepeda.

2.10 Analisis Statistik OLS

```

# Analisis statistik dengan OLS
import statsmodels.api as sm

X_train_const = sm.add_constant(X_train)
ols_model = sm.OLS(y_train, X_train_const).fit()
print(ols_model.summary())

```

Gambar 19. Analisis statistik dengan OLS

Kode:

```

import statsmodels.api as sm
X_train_const = sm.add_constant(X_train)
ols_model = sm.OLS(y_train, X_train_const).fit()
print(ols_model.summary())

```

Penjelasan:

- add_constant() menambahkan kolom konstanta.
- OLS() → menjalankan *Ordinary Least Squares Regression*.
- summary() → menampilkan hasil lengkap seperti:
 - Nilai koefisien
 - Nilai p (signifikansi variabel)
 - R^2 dan Adj. R^2
 - t-statistic

Output:

OLS Regression Results

Dep. Variable:

cnt

R-squared:

0.516

Model:

OLS

Adj. R-squared:

0.510

Method:

Least Squares

F-statistic:

76.70

Date:

Sun, 12 Oct 2025

Prob (F-statistic):

1.21e-85

Time:

00:57:33

Log-Likelihood:

-5030.1

No. Observations:

584

AIC:

1.008e+04

Df Residuals:

575

BIC:

1.012e+04

Df Model:

8

Covariance Type:

nonrobust

	coef	std err	t	P> t	[0.025	0.975]
const	2839.0904	400.244	7.093	0.000	2052.971	3625.210
temp	2455.6524	2244.785	1.094	0.274	-1953.325	6864.630
atemp	3491.0399	2540.096	1.374	0.170	-1497.959	8480.039
hum	-2146.3958	531.279	-4.040	0.000	-3189.880	-1102.912
windspeed	-2464.7207	786.875	-3.132	0.002	-4010.221	-919.220
season	419.2141	55.055	7.615	0.000	311.081	527.347
weathersit	-500.5628	132.909	-3.766	0.000	-761.609	-239.516
weekday	66.4468	27.612	2.406	0.016	12.214	120.679
workingday	154.6297	119.085	1.298	0.195	-79.265	388.525

Omnibus:

8.667

Durbin-Watson:

2.015

Prob(Omnibus):

0.013

Jarque-Bera (JB):

5.735

Skew:

0.081

Prob(JB):

0.0568

Kurtosis:

2.543

Cond. No.

286.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Gambar 20. Output dari analisis statistik dengan OLS

Penjelasan Output:

- Output tersebut menampilkan hasil regresi OLS (Ordinary Least Squares) secara lengkap.
- R-squared = 0.516 → model menjelaskan sekitar 51,6% variasi data.
- Adj. R-squared = 0.510 → menunjukkan tingkat penyesuaian model masih cukup baik.
- P>|t| menunjukkan signifikansi variabel:
 - Variabel hum, windspeed, season, weathersit, dan weekday memiliki p-value < 0.05, berarti berpengaruh signifikan terhadap jumlah penyewa sepeda (cnt).
 - Variabel temp, atemp, dan workingday tidak signifikan (p-value > 0.05).
- Const (Intercept) = 2839.09 → nilai dasar cnt saat semua variabel lain bernilai nol.

3. Kesimpulan

- 1) Model Multiple Linear Regression berhasil diterapkan untuk memprediksi jumlah penyewaan sepeda (cnt) berdasarkan kondisi cuaca dan hari.
- 2) Nilai $R^2 \approx 0.83$ menunjukkan model mampu menjelaskan sekitar 83% variasi data.
- 3) Variabel temp dan workingday berpengaruh positif terhadap jumlah penyewaan, sedangkan hum dan windspeed berpengaruh negatif.
- 4) Hasil OLS menunjukkan sebagian besar variabel signifikan secara statistik ($p < 0.05$).
- 5) Secara keseluruhan, algoritma regresi linear berganda ini bekerja cukup baik dan dapat digunakan untuk analisis prediksi jumlah penyewaan sepeda berdasarkan faktor lingkungan dan waktu.

Link Github Praktikum:

<https://github.com/rikaarahma/Machine-Learning/blob/main/praktikum03/notebook/praktikum03.ipynb>

Link Github Praktikum Mandiri:

<https://github.com/rikaarahma/Machine-Learning/blob/main/praktikum03/notebook/praktikum-mandiri03.ipynb>