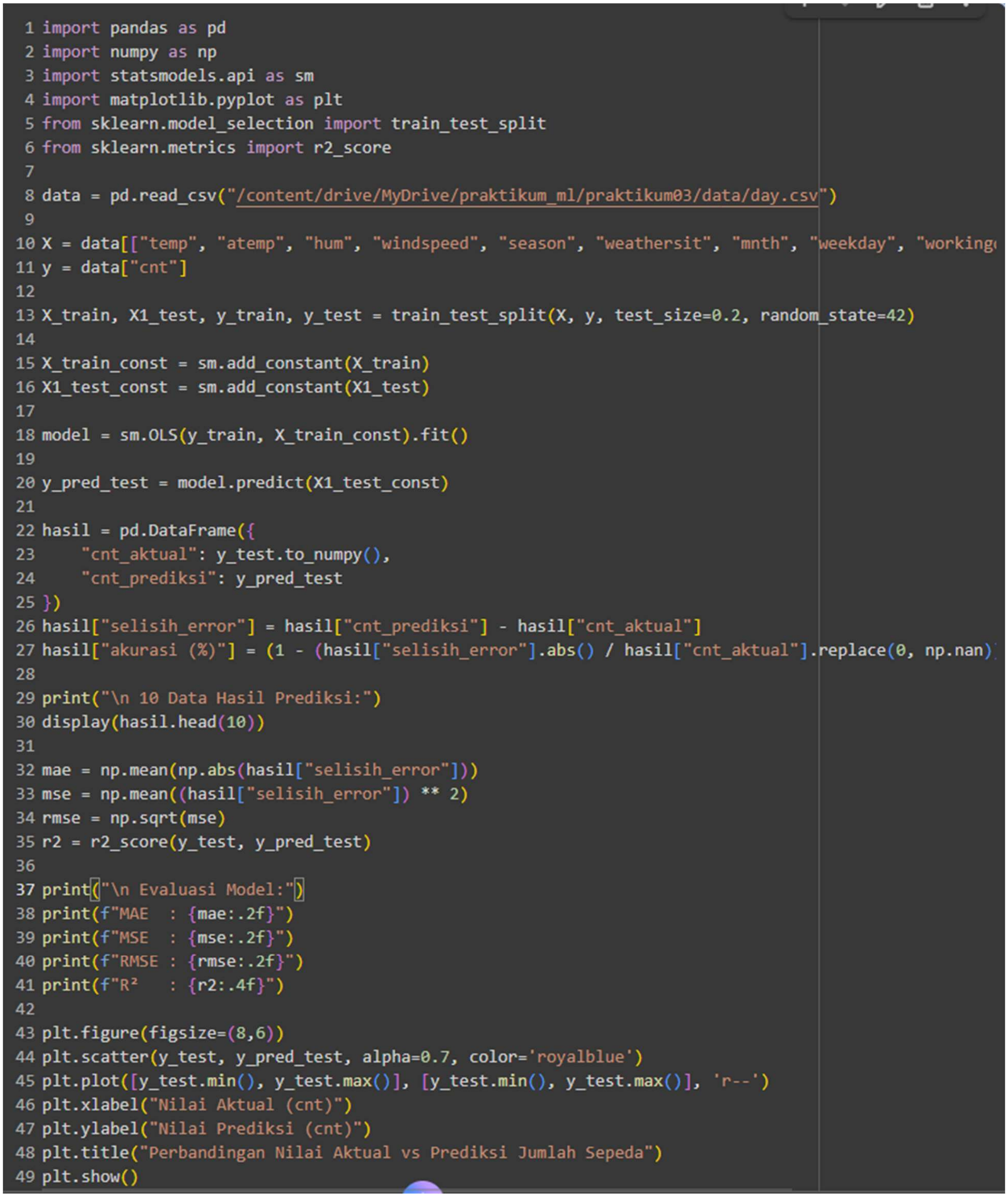
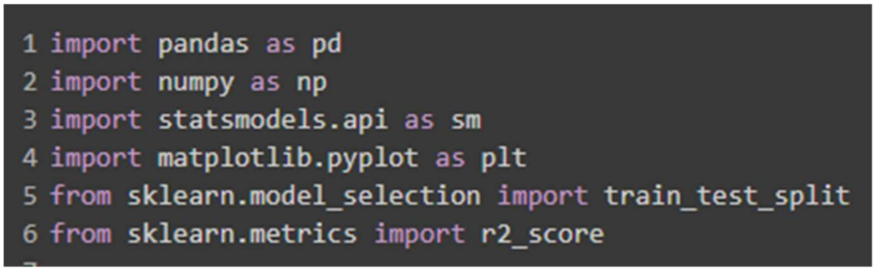
PRAKTIKUM 03

Keseluruhan Coding



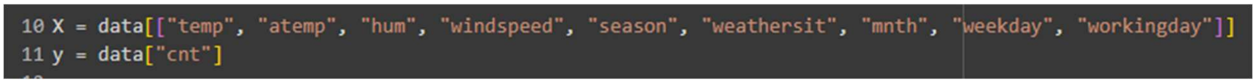
```python
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score

data = pd.read_csv("/content/drive/MyDrive/praktikum_ml/praktikum03/data/day.csv")

X = data[["temp", "atemp", "hum", "windspeed", "season", "weathersit", "mnth", "weekday", "working
y = data["cnt"]

X_train, X1_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

X_train_const = sm.add_constant(X_train)
X1_test_const = sm.add_constant(X1_test)

model = sm.OLS(y_train, X_train_const).fit()

y_pred_test = model.predict(X1_test_const)

hasil = pd.DataFrame({
    "cnt_aktual": y_test.to_numpy(),
    "cnt_prediksi": y_pred_test
})
hasil["selisih_error"] = hasil["cnt_prediksi"] - hasil["cnt_aktual"]
hasil["akurasi (%)"] = (1 - (hasil["selisih_error"].abs() / hasil["cnt_aktual"].replace(0, np.nan)

print("\n 10 Data Hasil Prediksi:")
display(hasil.head(10))

mae = np.mean(np.abs(hasil["selisih_error"]))
mse = np.mean((hasil["selisih_error"]) ** 2)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred_test)

print("\n Evaluasi Model:")
print(f"MAE  : {mae:.2f}")
print(f"MSE  : {mse:.2f}")
print(f"RMSE : {rmse:.2f}")
print(f"R²   : {r2:.4f}")

plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred_test, alpha=0.7, color='royalblue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel("Nilai Aktual (cnt)")
plt.ylabel("Nilai Prediksi (cnt)")
plt.title("Perbandingan Nilai Aktual vs Prediksi Jumlah Sepeda")
plt.show()
```

1. kode ini digunakan untuk menyiapkan semua pustaka yang diperlukan dalam proses analisis dan pemodelan regresi linear pada machine learning. Dengan library tersebut, data dapat diolah, dibuat model prediksi, serta dievaluasi hasilnya menggunakan metrik statistik.



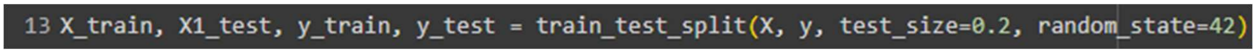
```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
```

2. Tentukan variable independent (X) dan (Y)



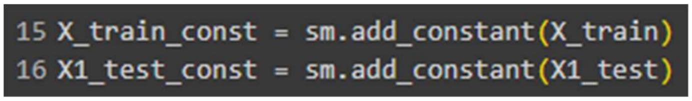
```
X = data[["temp", "atemp", "hum", "windspeed", "season", "weathersit", "mnth", "weekday", "workingday"]]
y = data["cnt"]
```

3. Bagi data menjadi training (80%) dan testing (20%)



```
X_train, X1_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

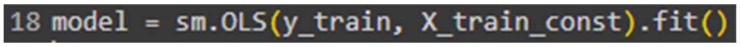
4. Tambahkan konstanta (intercept) untuk model statsmodels



```
X_train_const = sm.add_constant(X_train)
X1_test_const = sm.add_constant(X1_test)
```

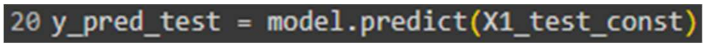
5. Latih model OLS

Berfungsi untuk mencari hubungan linear antara variabel independen (X) dan variabel dependen (Y) dengan cara meminimalkan selisih kuadrat antara nilai aktual dan nilai prediksi.



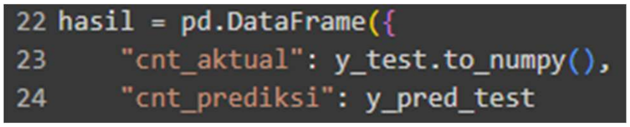
```
model = sm.OLS(y_train, X_train_const).fit()
```

6. Prediksi data yang uji



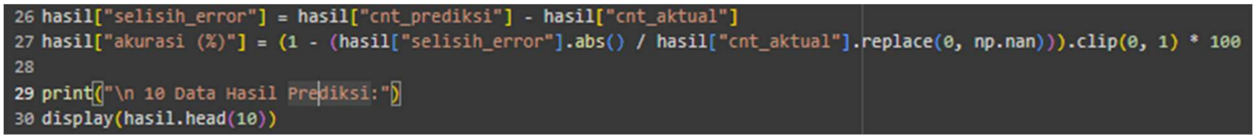
```
y_pred_test = model.predict(X1_test_const)
```

7. Membuat tabel hasil prediksi



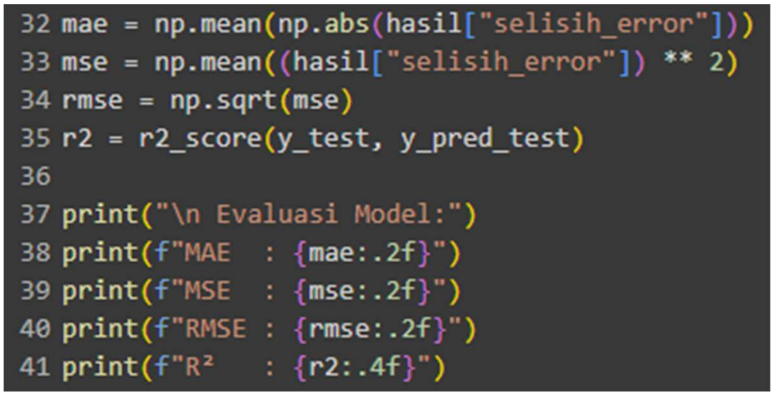
```
hasil = pd.DataFrame({
    "cnt_aktual": y_test.to_numpy(),
    "cnt_prediksi": y_pred_test
```

8. Hitung selisih error dan akurasi per baris untuk membandingkan hasil prediksi model dengan nilai aktual pada setiap baris data.



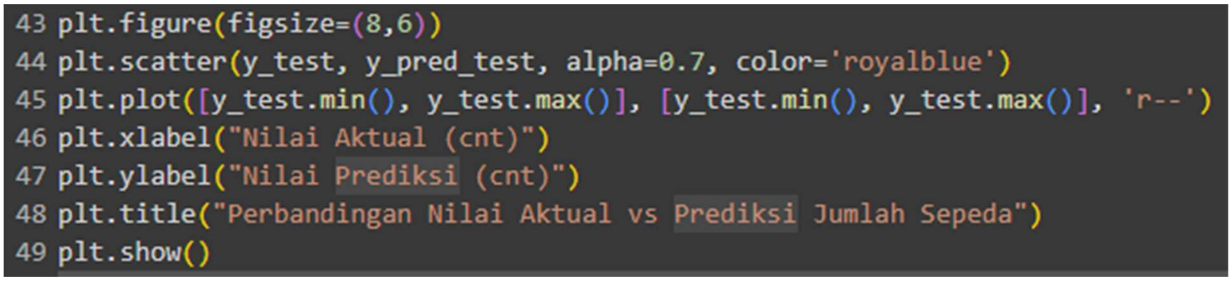
```
hasil["selisih_error"] = hasil["cnt_prediksi"] - hasil["cnt_aktual"]
hasil["akurasi (%)"] = (1 - (hasil["selisih_error"].abs() / hasil["cnt_aktual"].replace(0, np.nan))).clip(0, 1) * 100

print("\n 10 Data Hasil Prediksi:")
display(hasil.head(10))
```

9. Evaluasi Model untuk menilai seberapa baik kinerja model regresi dalam memprediksi data.



```
mae = np.mean(np.abs(hasil["selisih_error"]))
mse = np.mean((hasil["selisih_error"]) ** 2)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred_test)

print("\n Evaluasi Model:")
print(f"MAE  : {mae:.2f}")
print(f"MSE  : {mse:.2f}")
print(f"RMSE : {rmse:.2f}")
print(f"R²   : {r2:.4f}")
```

10. Tampilkan visualisasi hasil prediksi vs aktual



```
plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred_test, alpha=0.7, color='royalblue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel("Nilai Aktual (cnt)")
plt.ylabel("Nilai Prediksi (cnt)")
plt.title("Perbandingan Nilai Aktual vs Prediksi Jumlah Sepeda")
plt.show()
```