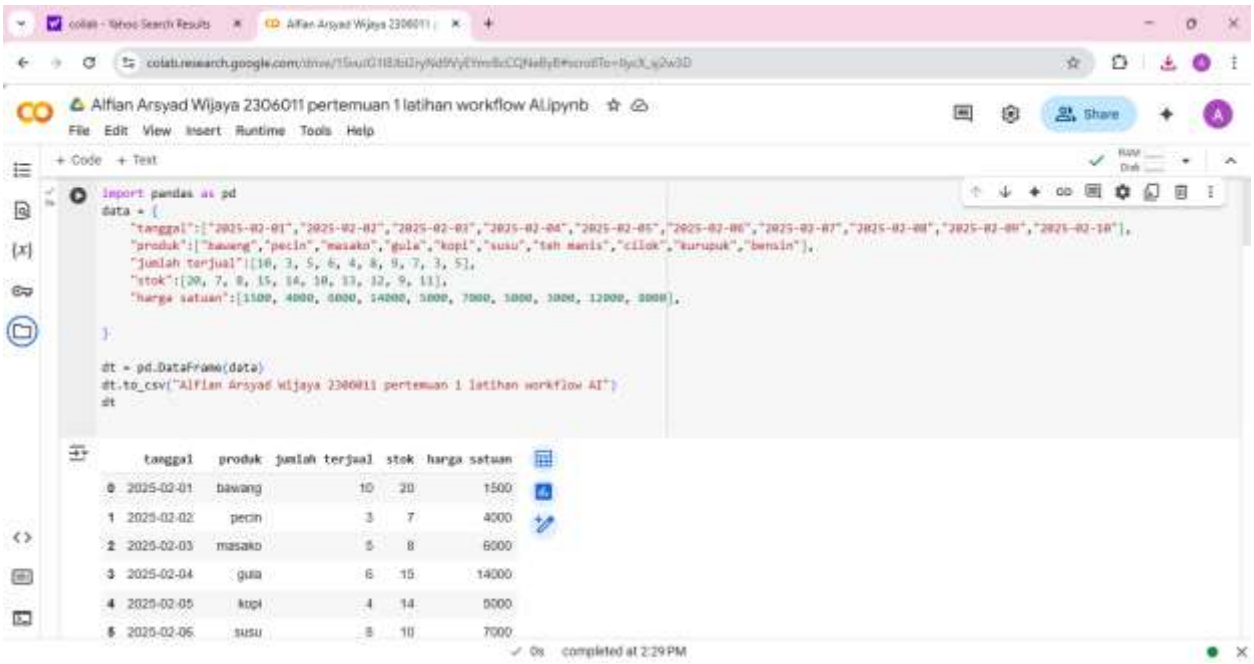
Alfian Arsyad Wijaya 2306011 pertemuan 1 latihan workflow AI.ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
import pandas as pd
data = {
"tanggal":["2025-02-01","2025-02-02","2025-02-03","2025-02-04","2025-02-05","2025-02-06","2025-02-07","2025-02-08","2025-02-09","2025-02-10"],
"produk":["bawang","pecin","masako","gula","kopi","susu","teh manis","cilok","kurupuk","bensin"],
"jumlah terjual":[10, 3, 5, 6, 4, 8, 9, 7, 3, 5],
"stok":[20, 7, 8, 15, 14, 10, 13, 12, 9, 11],
"harga satuan":[1500, 4000, 6000, 14000, 5000, 7000, 5000, 3000, 12000, 3000],
}
dt = pd.DataFrame(data)
dt.to_csv("Alfian Arsyad Wijaya 2306011 pertemuan 1 latihan workflow AI")
dt
tanggal produk jumlah terjual stok harga satuan
0 2025-02-01 bawang 10 20 1500
1 2025-02-02 pecin 3 7 4000
2 2025-02-03 masako 5 8 6000
3 2025-02-04 gula 6 15 14000
4 2025-02-05 kopi 4 14 5000
5 2025-02-06 susu 8 10 7000
0s completed at 2:29 PM


dt["tanggal"] = pd.to_datetime(dt["tanggal"])
dt.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
# Column Non-Null Count Dtype
0 tanggal 10 non-null datetime64[ns]
1 produk 10 non-null object
2 jumlah terjual 10 non-null int64
3 stok 10 non-null int64
4 harga satuan 10 non-null int64
dtypes: datetime64[ns](1), int64(3), object(1)
memory usage: 532.0+ bytes
[11] dt["tanggal"] = pd.to_datetime(dt["tanggal"])
dt["total penjualan"] = dt["jumlah terjual"] * dt["harga satuan"]
dt
tanggal produk jumlah terjual stok harga satuan total penjualan
0 2025-02-01 bawang 10 20 1500 15000
0s completed at 2:29 PM

```
dt["tanggal"] = pd.to_datetime(dt["tanggal"])

dt["total penjualan"] = dt["jumlah terjual"] * dt["harga satuan"]
dt
```

| | tanggal | produk | jumlah terjual | stok | harga satuan | total penjualan |
|---|---|---|---|---|---|---|
| 0 | 2025-02-01 | bawang | 10 | 20 | 1500 | 15000 |
| 1 | 2025-02-02 | pecin | 3 | 7 | 4000 | 12000 |
| 2 | 2025-02-03 | masako | 5 | 8 | 6000 | 30000 |
| 3 | 2025-02-04 | gula | 6 | 15 | 14000 | 84000 |
| 4 | 2025-02-05 | kopi | 4 | 14 | 5000 | 20000 |
| 5 | 2025-02-06 | susu | 8 | 10 | 7000 | 56000 |
| 6 | 2025-02-07 | teh manis | 9 | 13 | 5000 | 45000 |
| 7 | 2025-02-08 | cilok | 7 | 12 | 3000 | 21000 |
| 8 | 2025-02-09 | kurupuk | 3 | 9 | 12000 | 36000 |
| 9 | 2025-02-10 | bensin | 5 | 11 | 8000 | 40000 |

✓ 0s completed at 2:29 PM

```
dt["tanggal"] = pd.to_datetime(dt["tanggal"])

dt["total penjualan"] = dt["jumlah terjual"] * dt["harga satuan"]
dt["keutungan"] = dt["total penjualan"] * (dt["jumlah terjual"] * 100000)
dt
```

| | tanggal | produk | jumlah terjual | stok | harga satuan | total penjualan | keutungan |
|---|---|---|---|---|---|---|---|
| 0 | 2025-02-01 | bawang | 10 | 20 | 1500 | 15000 | 1500000000 |
| 1 | 2025-02-02 | pecin | 3 | 7 | 4000 | 12000 | 360000000 |
| 2 | 2025-02-03 | masako | 5 | 8 | 6000 | 30000 | 1500000000 |
| 3 | 2025-02-04 | gula | 6 | 15 | 14000 | 84000 | 5040000000 |
| 4 | 2025-02-05 | kopi | 4 | 14 | 5000 | 20000 | 800000000 |
| 5 | 2025-02-06 | susu | 8 | 10 | 7000 | 56000 | 4480000000 |
| 6 | 2025-02-07 | teh manis | 9 | 13 | 5000 | 45000 | 4050000000 |
| 7 | 2025-02-08 | cilok | 7 | 12 | 3000 | 21000 | 1470000000 |
| 8 | 2025-02-09 | kurupuk | 3 | 9 | 12000 | 36000 | 1080000000 |
| 9 | 2025-02-10 | bensin | 5 | 11 | 8000 | 40000 | 2000000000 |

✓ 0s completed at 2:29 PM

```
[14] from sklearn.model_selection import train_test_split
     from sklearn.tree import DecisionTreeClassifier
     from sklearn.metrics import accuracy_score

     X = dt[["jumlah terjual", "stok"]]
     Y = (dt["stok"] < 5).astype(int)

     X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

     model = DecisionTreeClassifier()
     model.fit(X_train, Y_train)
```

```
DecisionTreeClassifier()
```

```
[15] Y_pred = model.predict(X_test)
     print("akurasi model: ", accuracy_score(Y_test, Y_pred))
```

```
akurasi model:  1.0
```

```
produk_baru = [[8, 3]]
prediksi = model.predict(produk_baru)
```

✓ 0s completed at 2:29 PM


+ Code + Text
[14] model.fit(X_train, Y_train)
DecisionTreeClassifier
DecisionTreeClassifier()
[15] Y_pred = model.predict(X_test)
print("akurasi model: ", accuracy_score(Y_test, Y_pred))
akurasi model: 1.0
produk_baru = [[0, 3]]
prediksi = model.predict(produk_baru)
if prediksi [0] == 1:
print("produk perlu direstok!")
else:
print("stok masih cukup")
stok masih cukup
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted
warnings.warn(
0s completed at 2:29 PM