

Tugas Mandiri Dataframe I

1. Dari seluruh penumpang perempuan, sebutkan 5 nama penumpang yang umur termuda dan pada tahun berapa penumpang tersebut lahir.

Berikut jawabannya:

```
[7] # Menampilkan 5 nama penumpang dengan jenis kelamin Female(perempuan) dengan umur yang termuda diantara penumpang perempuan lainnya

Data_Female = df[df['payCardSex'] == 'F'].sort_values(by='payCardBirthDate', ascending=False).head(5)[['payCardName', 'payCardSex', 'payCardBirthDate']]
Data_Female
```

	payCardName	payCardSex	payCardBirthDate
4106	R. Lili Permata, S.E.	F	2012
20322	Catur Najmudin	F	2012
36527	Rahmi Saptono	F	2012
10122	Puspa Sihombing	F	2012
14649	Sutan Nasim Saragih	F	2012

Next steps: [View recommended plots](#)

2. Buatlah satu kolom baru umur yang berisi umur penumpang di tahun 2024 ini, kemudian tuliskan berapa jumlah transaksi yang dilakukan oleh penumpang yang sudah pensiun (>55 tahun), kemudian tuliskan juga jumlah masing-masing transaksi perempuan dan laki-laki.

Berikut jawabannya:

```
import pandas as pd

from datetime import datetime
# menambahkan satuan kolom umur
tahun_2024 = datetime.now().year
df["Age"] = tahun_2024 - df["payCardBirthDate"]

# menghitung jumlah transaksi penumpang
jumlah_pensiun = (df["Age"] > 55).sum()
jumlah_transaksi_laki_laki = df[(df["payCardSex"] == "M") & (df["Age"] > 55)].shape[0]
jumlah_transaksi_perempuan = df[(df["payCardSex"] == "F") & (df["Age"] > 55)].shape[0]

# cetak
print(df["Age"])
print(f"*100,\n",
      "\nJumlah transaksi oleh penumpang yang sudah pensiun (Age > 55 tahun):", jumlah_pensiun,
      "\nJumlah transaksi penumpang laki-laki:", jumlah_transaksi_laki_laki,
      "\nJumlah transaksi penumpang perempuan:", jumlah_transaksi_perempuan)
```

```
0      16
1      27
2      32
3      46
4      42
..
37895   20
37896   50
37897   48
37898   48
37899   39
Name: Age, Length: 37900, dtype: int64
```

```
=====

Jumlah transaksi oleh penumpang yang sudah pensiun (Age > 55 tahun): 2438
Jumlah transaksi penumpang laki-laki: 1783
Jumlah transaksi penumpang perempuan: 655
```

3. Tampilkan detail transaksi pertama berdasarkan tapInTime dengan menggunakan fungsi `iloc`.

Berikut jawabannya:

```
# Menampilkan data transaksi pertama dengan berpacu pada tapInTime menggunakan fungsi iloc
detail = df.sort_values(by = "tapInTime").iloc[0]
print(detail)

transID                                LDEA875J4U32YZ
payCardID                             213107623841273
payCardBank                           emoney
payCardName                           Natalia Pratama
payCardSex                             M
payCardBirthDate                       1964
corridorID                             11D
corridorName                           Pulo Gebang - Pulo Gadung 2 via PIK
direction                             0.0
tapInStops                             B05823P
tapInStopsName                         United Tractors 1
tapInStopsLat                          -6.18326
tapInStopsLon                          106.93243
stopStartSeq                           28
tapInTime                             2023-04-01 06:22:27
tapOutStops                            B03090P
tapOutStopsName                       Raya Penggilingan
tapOutStopsLat                         -6.183068
tapOutStopsLon                         106.93194
stopEndSeq                             29.0
tapOutTime                             2023-04-01 08:15:45
payAmount                             3500.0
umur                                   60
jumlah_pemberhentian                   1.0
Name: 37751, dtype: object
```

4. Ada berapa transaksi gratis (0 rupiah) yang terdapat dalam data? Siapakah penumpang yang melakukan transaksi perjalanan dengan yang melewati jumlah pemberhentian terbanyak (stopEndSeq - stopStartSeq) dan sebutkan rute perjalanan penumpang tersebut?

Berikut jawabannya:

```
# temukan transaksi gratis (0 rupiah)
transaksi_gratis= (df['payAmount'] == 0).sum()

# temukan penumpang yang melakukan transaksi perjalanan dengan jumlah pemberhentian terbanyak
df['Banyaknya_pemberhentian'] = df['stopEndSeq'] - df['stopStartSeq']
sortir_pemberhentian = df.sort_values(by='Banyaknya_pemberhentian', ascending=False)
transaksi_pemberhentian_terbanyak = sortir_pemberhentian.iloc[0]

# cetak hasil
print("Jumlah banyaknya transaksi gratis =",transaksi_gratis)
print("Informasi tentang penumpang dengan transaksi paling banyak pemberhentian:")
print("Nama Penumpang:", transaksi_pemberhentian_terbanyak["payCardName"])
print("Rute Perjalanan:", [transaksi_pemberhentian_terbanyak["tapInStopsName"],transaksi_pemberhentian_terbanyak["tapOutStopsName"]])
print("Total Pemberhentian:",transaksi_pemberhentian_terbanyak["Banyaknya_pemberhentian"])

Jumlah banyaknya transaksi gratis = 16648
Informasi tentang penumpang dengan transaksi paling banyak pemberhentian:
Nama Penumpang: Karen Pertiwi
Rute Perjalanan: ['RS Pelabuhan Jakarta', 'Kel. Rorotan']
Total Pemberhentian: 64.0
```

5. Berapakah biaya yang dikeluarkan dan rute perjalanan untuk transaksi dengan jarak tempuh terlama dan jumlah pemberhentian terbanyak?

Berikut jawabannya:

```
[21] # temukan transaksi dengan jarak tempuh terlama dengan menggunakan kolom tapInTime dan tapOutTime untuk mengetahui waktu terlama
df["tapInTime"] = pd.to_datetime(df["tapInTime"])
df["tapOutTime"] = pd.to_datetime(df["tapOutTime"])
df["durasi"] = (df["tapOutTime"] - df["tapInTime"]).dt.total_seconds()

# hitung jumlah pemberhentian
df["jumlah_pemberhentian"] = df["stopEndSeq"] - df["stopStartSeq"]

# mensortir berdasarkan kolom durasi(jarak tempuh terlama) dan jumlah pemberhentian
df = df.sort_values (by=["durasi", "jumlah_pemberhentian"], ascending = [False, False]).reset_index(drop=True)
jumlah_pemberhentian = df.loc[0, "jumlah_pemberhentian"]
rute_terlama = [df.loc[0, "tapInStopsName"], df.loc[0, "tapOutStopsName"]]
biaya = df.loc[0, "payAmount"]

# cetak hasil
print("Biaya perjalanan untuk transaksi dengan jarak tempuh terlama yang dikeluarkan sebanyak:", biaya, "rupiah")
print("Rute perjalanan untuk transaksi dengan jarak tempuh terlama:", rute_terlama)
print("Rute perjalanan untuk transaksi dengan jumlah pemberhentian terbanyak :", jumlah_pemberhentian, "kali pemberhentian")

Biaya perjalanan untuk transaksi dengan jarak tempuh terlama yang dikeluarkan sebanyak: 3500.0 rupiah
Rute perjalanan untuk transaksi dengan jarak tempuh terlama: ['Sbr. Jln. H. Ung', 'Jembatan Item']
Rute perjalanan untuk transaksi dengan jumlah pemberhentian terbanyak : 12.0 kali pemberhentian
```