

**Kelompok 7: BATCH 41 DS**

1. Harianto
2. Wika Rabila Putri

```
import pandas as pd
import numpy as np
```

```
# Originally created by Harianto
# 25 Januari 2024
# 14:47 WIB
```

```
from google.colab import drive
drive.mount('/content/drive')
%cd /content/drive/My Drive/Colab Notebooks/RAKAMIN PERSONAL/learning/Homework Loop and Iteration
```

```
Mounted at /content/drive
/content/drive/My Drive/Colab Notebooks/RAKAMIN PERSONAL/learning/Homework Loop and Iteration
```

1. Tentukan dari variabel `list_plat_mobil` = ['B 1234 AB', 'B 6721 TY', 'B 1233 AY', 'B 6629 DD', 'B 1111 AM', 'B 6726 D', 'D 11223 KJ', 'AE 44677 GH', 'AE 67269 AA'] Ada berapa jumlah plat genap dan ganjil.

```
#1 ANSWER
list_plat_mobil = ['B 1234 AB', 'B 6721 TY', 'B 1233 AY', 'B 6629 DD', 'B 1111 AM', 'B 6726 D', 'D 11223 KJ', 'AE 44677 GH', 'AE 67269 /
results = ['GENAP' if int(i.split(' ')[1])%2 == 0 else 'GANJIL' for i in list_plat_mobil]
ganjil = results.count('GANJIL')
genap = results.count('GENAP')

print("Jumlah plat nomor ganjil :", ganjil, "Jumlah plat nomor genap :", genap)
print(results)

Jumlah plat nomor ganjil : 7 Jumlah plat nomor genap : 2
['GENAP', 'GANJIL', 'GANJIL', 'GANJIL', 'GANJIL', 'GENAP', 'GANJIL', 'GANJIL', 'GANJIL']
```

2. Buatlah deret fibonacci 12 angka yang dimulai dari 2 dan 3 menggunakan iterasi cara for atau while. Output yang diharapkan adalah [2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377]

```
#2 ANSWER
a, b = 2, 3
fibonacci_series = [a, b]
n = 10
for _ in range(n):
    a, b = b, a + b
    fibonacci_series.append(b)
print(fibonacci_series)

[2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377]
```

3. Team Campaign PT ABC akan membuat promo bundling category product (berisi 3 category setiap bundling) untuk promo 17 Agustus. PT ABC memiliki 10 category product. Ketentuan Bundling:

1. Tidak boleh hanya memiliki 1 jenis category dalam 1 bundling
2. 1 Bundling boleh memiliki 2 category yang sama
3. Urutan tidak menjadi faktor pembeda ( $c1 - c2 - c3 = c2 - c1 - c3$ ).

Tentukan ada berapa jumlah kemungkinan bundling yang bisa terbentuk? Berikut list code category `list_id_category` = [c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]

#3 ANSWER

```
list_id_category = ['c1', 'c2', 'c3', 'c4', 'c5', 'c6', 'c7', 'c8', 'c9', 'c10']
basket = []

for category1 in list_id_category:
    for category2 in list_id_category:
        if category1 != category2 and category2 != category1:
            for category3 in list_id_category:
                if category3 != category2 and category3 != category1:
                    combination = sorted([category1, category2, category3])
                    combination = combination[0]+'-'+combination[1]+'-'+combination[2]
                    if combination not in basket:
                        basket.append(combination)

print("Jumlah kombinasi :", len(basket))
print("Kombinasi: ", basket)

Jumlah kombinasi : 120
Kombinasi: ['c1-c2-c3', 'c1-c2-c4', 'c1-c2-c5', 'c1-c2-c6', 'c1-c2-c7', 'c1-c2-c8', 'c1-c2-c9', 'c1-c10-c2', 'c1-c3-c4', 'c1-c3-c5'
```

4. Buatlah sebuah function string\_processor dengan inputan 2 parameter, yaitu suatu string 1 kata dan nama proses yang akan dilakukan. Function ini memiliki 4 proses:

1. first\_char : return huruf pertama dari string yang diinput
2. last\_char : return huruf terakhir dari string yang diinput
3. num\_chars : return jumlah huruf string input
4. num\_vowels : return jumlah huruf vocal pada string input

#4 ANSWER

```
def string_processor(text :str, method : str):
    if method.lower() == 'first_char':
        return text[0]
    elif method.lower() == 'last_char':
        return text[-1]
    elif method.lower() == 'num_char':
        return len(text)
    elif method.lower() == 'num_vowels':
        return sum([1 for i in text if i in ['a','e','i','o','u']])
    else:
        return 'method tersedia : first_char, last_char, num_char & num_vowels'

print(string_processor("Harianto","first_char"))
print(string_processor("Harianto","last_char"))
print(string_processor("Harianto", "num_char"))
print(string_processor("Harianto","num_vowels"))
```

H  
o  
8  
4

5. Menggunakan dataset rakamin\_store\_city\_monthly! Buatlah 1 kolom baru pada dengan teknik apply lambda tanpa function def (cukup one liner) dengan nama kolom is\_gmail dengan value True jika menggunakan gmail dan False jika selain gmail. Tentukan ada berapa user yang menggunakan email gmail pada tahun 2018!

#5 ANSWER

```
df = pd.read_csv('rakamin_store_city_monthly.csv', delimiter=';')
df['is_gmail'] = df.apply(lambda x: x['email'].endswith('gmail.com'), axis=1)
filtered = df[(df['month'].str[:4] == '2018') & (df['is_gmail'] == True)]
print("Jumlah pengguna dengan provider email GMAIL pada tahun 2018:", len(filtered))
filtered.head()
```

Jumlah pengguna dengan provider email GMAIL pada tahun 2018: 3

	month	user_id	username	email	kota	total_visit	total_orde
57	2018-02	20	Yutama Riahdita	yutamariahdita@gmail.com	Jakarta Barat	21	3
61	2018-03	28	Fahmi Haq	fahmihq@gmail.com	Depok	124	3

6. Menggunakan dataset superstore! Buatlah function yang bisa menghitung Jumlah Sales dan Profit pada suatu tahun dan region tertentu dengan default tahun 2017. Tentukan total sales dan profit:

1. South Region pada tahun 2016
2. West Region pada tahun 2017

#6 ANSWER

```
data = pd.read_excel('superstore.xls')
data['order_year'] = data['Order Date'].astype(str).str[:4] #Biar mudah

def provit_and_sales_counter(regional: str = "South", year : int = 2017):
    df = data.groupby(['Region','order_year']).agg({'Sales' : 'sum', 'Profit' : 'sum'}).reset_index()
    df = df[(df['Region'].astype(str).str.lower() == regional.lower()) & (df['order_year'].astype(int) == year)]
    return df
print("South Region pada tahun 2016")
print(provit_and_sales_counter('South',2016))
print("West Region pada tahun 2017")
print(provit_and_sales_counter('West'))
```

```
South Region pada tahun 2016
  Region order_year      Sales      Profit
10  South      2016  93610.2235  17702.8084
West Region pada tahun 2017
  Region order_year      Sales      Profit
15  West      2017  250128.3655  43808.9561
```

7.Menggunakan dataset superstore! Buatlah 3 kelompok 'segment' (high, medium, dan low) berdasarkan performance Sales yang digunakan sebagai acuan tim bisnis untuk mengelompokkan customer superstore. Instruksi:

1. Bantulah tim bisnis untuk menentukan kriteria pengelompokan untuk 3 segmentasi customer ini, dan gunakan iterrows atau apply lambda sebagai teknik pembuatan kolom baru!
2. Berikan alasan kalian dalam penentuan kriteria pengelompokan customer yang dibuat!
3. Tentukan jumlah customer pada masing-masing segment di tahun 2017!

# 7 ANSWER

```
# Dengan dua Group By ['Customer ID','order_year']

# CREATE ORDER YEAR COLUMN
# This customer dataset already have order_year, we add it in previous answer

# AGGREGATION CUSTOMER DATA BEFORE DETERMINE SEGMENTATION
customer = data.groupby(['Customer ID','order_year']).agg({'Sales' : 'sum', 'Order ID' : 'count'}).reset_index()

#FIND DESCRIPTIVE STATISTIC BY SALES COLUMN
descriptive = customer['Sales'].describe()
batas_low, batas_medium , batas_high= descriptive['25%'], descriptive['50%'], descriptive['75%']
print({'parameter' : {'batas_low' : batas_low, 'batas_medium' : batas_medium, 'batas_high' : batas_high}})

#MAKE FUNCTION TO DETERMINE CUSTOMER SEGMENTATION
def kategori_penjualan(nilai):
    if nilai <= batas_low:
        return 'Low'
    elif nilai <= batas_medium:
        return 'Medium'
    else:
        return 'High'

#APPLY SEGMENTATION TO DATASET BY USING LAMBDA
customer['Segment'] = customer.apply(lambda x: kategori_penjualan(x['Sales']), axis=1)
customer.columns = ['Customer ID', 'order_year', 'Total Sales','Total Transaction', 'Customer Segment']
customer
# FILTER BY ORDER YEAR = 2017
final = customer[customer['order_year'] == '2017']

#FIND NUM OF CUSTOMER BASED ON SEGMENT IN 2017
final.groupby('Customer Segment').agg({'Customer ID': 'count'}).reset_index()
```

```
{'parameter': {'batas_low': 120.707, 'batas_medium': 472.0440000000001, 'batas_high':
```

	Customer ID	order_year	Total Sales	Total Transaction	Customer Segment
0	AA-10315	2014	756.048	4	High
1	AA-10315	2015	26.960	1	Low
2	AA-10315	2016	4406.072	4	High
3	AA-10315	2017	374.480	2	Medium

```
# 7 ANSWER
```

```
# Dengan satu Group By ['Customer ID']
```

```
# CREATE ORDER YEAR COLUMN
```

```
# This customer dataset already have order_year, we add it in previous answer
```

```
# AGGREGATION CUSTOMER DATA BEFORE DETERMINE SEGMENTATION
```

```
customers = data[data['order_year'] == '2017']
```

```
customers = data.groupby(['Customer ID']).agg({'Sales' : 'sum', 'Order ID' : 'count'}).reset_index()
```

```
#FIND DESCRIPTIVE STATISTIC BY SALES COLUMN
```

```
descriptive = customers['Sales'].describe()
```

```
batas_low, batas_medium , batas_high= descriptive['25%'], descriptive['50%'], descriptive['75%']
```

```
print({'parameter' : {'batas_low' : batas_low, 'batas_medium' : batas_medium, 'batas_high' : batas_high}})
```

```
#MAKE FUNCTION TO DETERMINE CUSTOMER SEGMENTATION
```

```
def kategori_penjualan(nilai):
```

```
    if nilai <= batas_low:
```

```
        return 'Low'
```

```
    elif nilai <= batas_medium:
```

```
        return 'Medium'
```

```
    else:
```

```
        return 'High'
```

```
#APPLY SEGMENTATION TO DATASET BY USING LAMBDA
```

```
customers['Segment'] = customers.apply(lambda x: kategori_penjualan(x['Sales']), axis=1)
```

```
customers.columns = ['Customer ID', 'Total Sales', 'Total Transaction', 'Customer Segment']
```

```
# FILTER BY ORDER YEAR = 2017
```

```
#FIND NUM OF CUSTOMER BASED ON SEGMENT IN 2017
```

```
customers.groupby('Customer Segment').agg({'Customer ID': 'count'}).reset_index()
```

```
# final = data.merge(customers, how='left', left_on=['Customer ID'], right_on=['Customer ID'])
```

```
{'parameter': {'batas_low': 1146.05, 'batas_medium': 2256.3940000000002, 'batas_high': 3785.276}}
```

	Customer Segment	Customer ID
0	High	396
1	Low	199
2	Medium	198

```
#8
```

```
# Buatlah suatu module bebas, dengan suatu manfaat tertentu (Belum ada pada materi Function).
```

```
# Komponen yang wajib ada :
```

```
# -Function-Docstring-Memanfaatkan conditional statement / iterasi (at least menggunakan salah satunya)
```

```
# -Terdapat kondisi handling exception error
```

```
# HERE IS THE FILE #
```

```
# https://drive.google.com/drive/folders/1izwtAQ4VdYXfbk6zJdqKjoVwvGBv0QTf?usp=drive_link
```

```
# FOLDER rakamin_homework_module
```

```
# OBJECTIVE :
```

```
# 1 FACE COMPARISON
```

```
# 2 LOOP SQUARE AND TRIANGLE
```