

- Nama: Ferrent Georgina Rini Tacoh
- Kelas DSF.4: Data Science

Import Library Pandas dan Numpy

```
In [26]: import pandas as pd  
import numpy as np
```

Question 1

Maximum and Minimum only gives us 1 value, which is the largest and smallest value of a column.
Can you find a way to give us the 3rd largest value and 5th smallest value in a column?

```
In [27]: df = pd.DataFrame({  
    'Name': ['Aldo', 'Bryan', 'Cindy', 'Dilan', 'Erika', 'Fitri', 'Gerald', 'Halim'],  
    'Score': [80, 85, 87, 75, 77, 83, 79, 83]  
})
```

```
In [28]: df
```

```
Out[28]:
```

	Name	Score
0	Aldo	80
1	Bryan	85
2	Cindy	87
3	Dilan	75
4	Erika	77
5	Fitri	83
6	Gerald	79
7	Halim	83

```
In [29]: df['Score'].max()
```

```
Out[29]: 87
```

`df['Score'].max()` gives us the highest score. What should we do if we want to get the 3rd highest score?

```
In [30]: df[df['Score'] == df['Score'].nlargest(3).iloc[-1]]
```

```
Out[30]:
```

	Name	Score
5	Fitri	83
7	Halim	83

```
In [31]: df[df['Score'] == df['Score'].nsmallest(5).iloc[-1]]
```

```
Out[31]:
```

	Name	Score
5	Fitri	83
7	Halim	83

```
In [31]:
```

Question 2

Cummulative Sum. Example:

```
In [32]: df_2 = pd.DataFrame({  
    'Single Value':[0,1,3,4,7]  
})
```

```
In [33]: df_2
```

```
Out[33]:
```

	Single Value
0	0
1	1
2	3
3	4
4	7

How to add another column so that we get the cummulative sum for each row?

```
In [34]: df_2['Cummulative'] = df_2['Single Value'].cumsum()  
df_2
```

```
Out[34]:
```

	Single Value	Cummulative
0	0	0
1	1	1
2	3	4
3	4	8
4	7	15

Question 3

Write an if..else... code in Python to check whether a word is a palindrome or not.

Input: 'malam'

Output: 'malam' is a Palindrome

Input: 'ABBA'

Output: 'ABBA' is a Palindrome

Input: 'Python'

Output: 'Python' is not a Palindrome.

```
In [35]: def checkPal(text):  
        if (text.lower() == text[::-1].lower()):  
            print("'" + text + "' is a palindrome")  
        else:  
            print("'" + text + "' is not a palindrome")
```

```
In [36]: checkPal('malam')
```

'malam' is a palindrome

```
In [37]: checkPal('ABBA')
```

'ABBA' is a palindrome

```
In [38]: checkPal('Python')
```

'Python' is not a palindrome

Question 4

As seen in the powerpoint slide, how to change the current dataframe into the desired result?

```
In [39]: df_4 = pd.DataFrame({
    'Buyer': ['Aldo', 'Budi', 'Cindy', 'Dilan'],
    'Items Bought': [['book', 'magazine'], ['realme 8', 'screen protector'],
                     ['iPhone 11', 'power bank'], ['face shield', 'mask', 'hand sanitizer']]
})
```

```
In [40]: df_4
```

```
Out[40]:
```

	Buyer	Items Bought
0	Aldo	[book, magazine]
1	Budi	[realme 8, screen protector]
2	Cindy	[iPhone 11, power bank]
3	Dilan	[face shield, mask, hand sanitizer]

```
In [41]: # prepare an empty list for each columns
buyer_list = []
item_list = []

# generate each row entry as a list
for i in range(0, len(df_4)):
    for item in df_4['Items Bought'][i]:
        buyer_list.append(df_4['Buyer'][i])
        item_list.append(item)

# create a new df from the generated list
df_4_long = pd.DataFrame(zip(buyer_list, item_list))
df_4_long.columns = df_4.columns

df_4_long
```

```
Out[41]:
```

	Buyer	Items Bought
0	Aldo	book
1	Aldo	magazine
2	Budi	realme 8
3	Budi	screen protector
4	Cindy	iPhone 11
5	Cindy	power bank
6	Dilan	face shield
7	Dilan	mask
8	Dilan	hand sanitizer

Question 5

If item 1 and item 2 shows the same product when switched, then it is counted as a duplicate. For examples are row 0 and 2, row 6 and row 7.

How to delete the duplicate?

```
In [42]: df_5 = pd.DataFrame({
    'Item 1': ['Buku', 'Buku', 'Majalah', 'iPhone', 'iPhone', 'Alat Tulis', 'Mouse'],
    'Item 2': ['Majalah', 'Koran', 'Buku', 'Charger', 'Power Bank', 'Buku', 'Keyboard']
})
```

```
In [43]: df_5
```

```
Out[43]:
```

	Item 1	Item 2
0	Buku	Majalah
1	Buku	Koran
2	Majalah	Buku
3	iPhone	Charger
4	iPhone	Power Bank
5	Alat Tulis	Buku
6	Mouse	Keyboard
7	Keyboard	Mouse

```
In [44]: original_tuple = [(item_1, item_2) for item_1, item_2 in zip(df_5['Item 1'], df_5['Item 2'])
    original_tuple
```

```
Out[44]: [('Buku', 'Majalah'),
 ('Buku', 'Koran'),
 ('Majalah', 'Buku'),
 ('iPhone', 'Charger'),
 ('iPhone', 'Power Bank'),
 ('Alat Tulis', 'Buku'),
 ('Mouse', 'Keyboard'),
 ('Keyboard', 'Mouse')]
```

```

In [45]: # prepare list to keep track for deduplicated entries and the duplicates
dedup_list = []
duplicate_list = []

for tup in original_tuple:
    # condition for duplicates: just ignore
    if tup in duplicate_list:
        pass
    # condition for the 1st occurrence of tuple with duplicates
    # i.e. the flipped version is in the original list, but we haven't appended
    elif tup[::-1] in original_tuple and tup not in duplicate_list:
        # append the original tuple to the deduplicated list
        dedup_list.append(tup)
        # append the flipped version to the list of duplicates
        duplicate_list.append(tup[::-1])
    # condition for tuples with no duplicates: just append to the deduplicated list
    else:
        dedup_list.append(tup)

# create a new DF based on the deduplicated list
df_5_dedup = pd.DataFrame(dedup_list)
df_5_dedup.columns = df_5.columns

```

```

In [46]: df_5_dedup

```

```

Out[46]:

```

	Item 1	Item 2
0	Buku	Majalah
1	Buku	Koran
2	iPhone	Charger
3	iPhone	Power Bank
4	Alat Tulis	Buku
5	Mouse	Keyboard