Hi All!

M Ammar Fauzan

Data Manager

EVERMOS

Sistem Informasi
(2014-2018)

Data Scientist
(Sept 2018- August 2020)

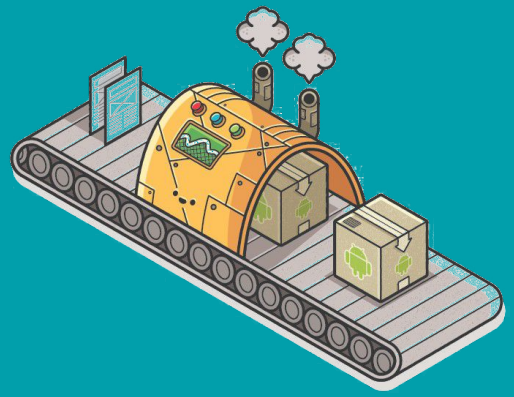
Business Intelligence .Mgr
(Sept 2020- Sept 2022)

Data Manager
(Sept 2022- Present)

Muhammad Ammar Fauzan
https://www.linkedin.com/in/muhammad-ammar-fauzan-748883117/

Rakamin Academy

Module Function Sederhana



“ Pada beberapa kasus kita membutuhkan untuk menyimpan function pada file terpisah dari project jupyter. Hal ini dilakukan agar project menjadi lebih simple dan rapi. ”

Snippet Code Import Function

Contoh Code:

```
1  from myfunction import calculator
```

```
1  calculator(x = 10, y= 20, metode='perkalian')
```

```
200
```

Snippet Code Import Function

myfunction.py

```python
def calculator(x, y, metode='penjumlahan'):
    """
    Objective
    Function Calculator for basic artimatic 2 variable number

    Arguments:
    - x = first number, type integer
    - y = second number, type integer
    - metode = (default 'penjumlahan'),
        expected values: [penjumlahan, pengurangan, pembagian, perkalian, pangkat]

    """

    metode = metode.lower()
    if metode == 'penjumlahan':
        result = x+y
    elif metode == 'pengurangan':
        result = x-y
    elif metode == 'pembagian':
        result = x/y
    elif metode == 'perkalian':
        result = x*y
    elif metode == 'pangkat':
        result = x**y
    else:
        result = '-'
        print('Inputan tidak sesuai!')
    return result
```

Step by step

1. Membuat kode program function pada suatu file **berformat .py** di Python Script Editor

2. file disimpan 1 folder atau folder berbeda dengan project .ipynb

Snippet Code Import Function

myfunction.py

```python
def calculator(x, y, metode='penjumlahan'):
    """
    Objective
    Function Calculator for basic artimatic 2 variable number

    Arguments:
    - x = first number, type integer
    - y = second number, type integer
    - metode = (default 'penjumlahan'),
        expected values: [penjumlahan, pengurangan, pembagian, perkalian, pangkat]

    """

    metode = metode.lower()
    if metode == 'penjumlahan':
        result = x+y
    elif metode == 'pengurangan':
        result = x-y
    elif metode == 'pembagian':
        result = x/y
    elif metode == 'perkalian':
        result = x*y
    elif metode == 'pangkat':
        result = x**y
    else:
        result = '-'
        print('Inputan tidak sesuai!')
    return result
```

Step by step

1. Membuat kode program function pada suatu file **berformat .py** di Python Script Editor

2. file disimpan 1 folder atau folder berbeda dengan project .ipynb

3. Tambahkan **DocStrings** pada statement pertama header line function. Berfungsi untuk memudahkan memahami context function saat akan digunakan.

Manfaat Docstrings

```
Signature: calculator(x, y, metode='penjumlahan')
Docstring:
Objective
Function Calculator for basic artimatic 2 variable number

Arguments:
- x = first number, type integer
- y = second number, type integer
- metode = (default 'penjumlahan'),
    expected values: [penjumlahan, pengurangan, pembagian,
perkalian, pangkat]
File:
c:\users\user\evdata\project\rakamin\myfunction.py
Type:      function
```

Manfaat Docstring untuk memudahkan memahami penggunaan function

Cara mengimport Function

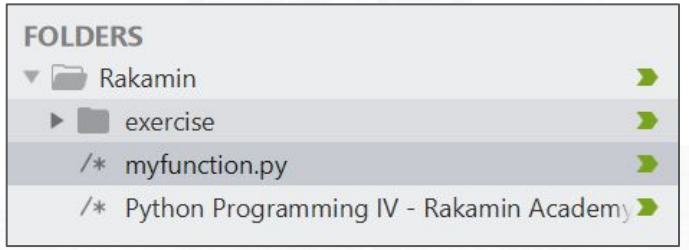
Contoh Code:

1

```
1  from myfunction import calculator
```

```
1  calculator(x = 10, y= 20, metode='perkalian')
```

```
200
```



Path
from <nama file> import <nama function>

Cara mengimport Function

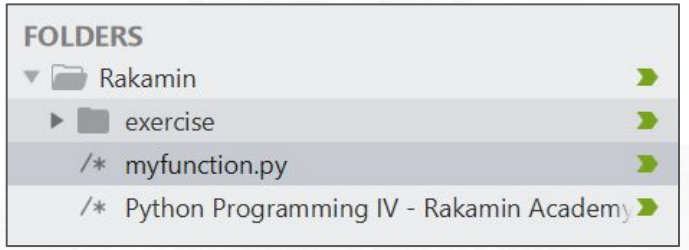
Contoh Code:

1

```
from myfunction import calculator
```

```
calculator(x = 10, y= 20, metode='perkalian')
```

```
200
```



Path
from <nama file> import <nama function>

2

```
import myfunction as mf
```

```
mf.calculator(x = 10, y= 20, metode='perkalian')
```

```
200
```

Path
import <nama file>

Memungkinkan untuk berbeda directory #1

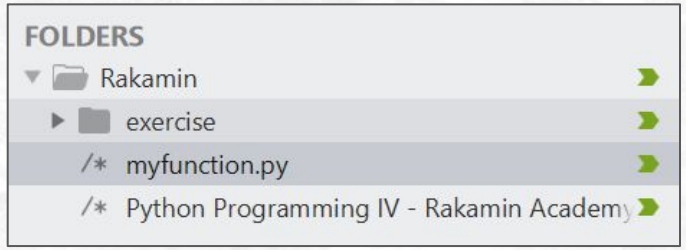
Contoh Code:

```
1  from exercise.myfunction_special import calculator
```

```
1  calculator(x = 10, y= 20, metode='penjumlahan')
```

```
30
```

Path
from <nama folder1>.<nama file> import <nama function>



Memungkinkan untuk berbeda directory #2

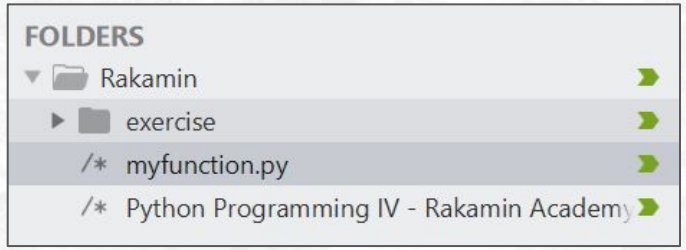
Contoh Code:

```
1  import exercise.myfunction_special as ms
```

```
1  ms.calculator(x = 10, y= 20, metode='penjumlahan')
```

```
30
```

Path
import <nama folder1>.<nama file>



Memungkinkan untuk berbeda directory #3

Contoh Code:

```
1  import sys
2  sys.path.append('C:/Users/User/EVDATA/Project')
```

```
1  import myfunction_exercise
```

```
1  myfunction_exercise.calculator(x = 10, y= 20, metode='penjumlahan')
```

```
30
```

Step by Step

- **Menggunakan package sys (path dan append) → Untuk membuat terhubung dengan directory tempat module telah dibuat**
- **import <Nama Function>**

Lebih dari 1 Function dalam 1 File

Contoh Code:

myfunction.py

```
1   def calculator(x, y, metode='penjumlahan'):
28
29  def calculator_super(x, y, z=None, metode='penjumlahan'):
57
```

```
1   import myfunction as mf
```

```
1   mf.calculator(10, 20, 30)
```

```
60
```

f calculator function

f calculator_super function

Contoh case implementasi module

Contoh Code:

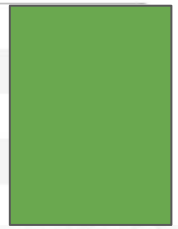
```python
from connectEVM import queryR
import pandas as pd
```

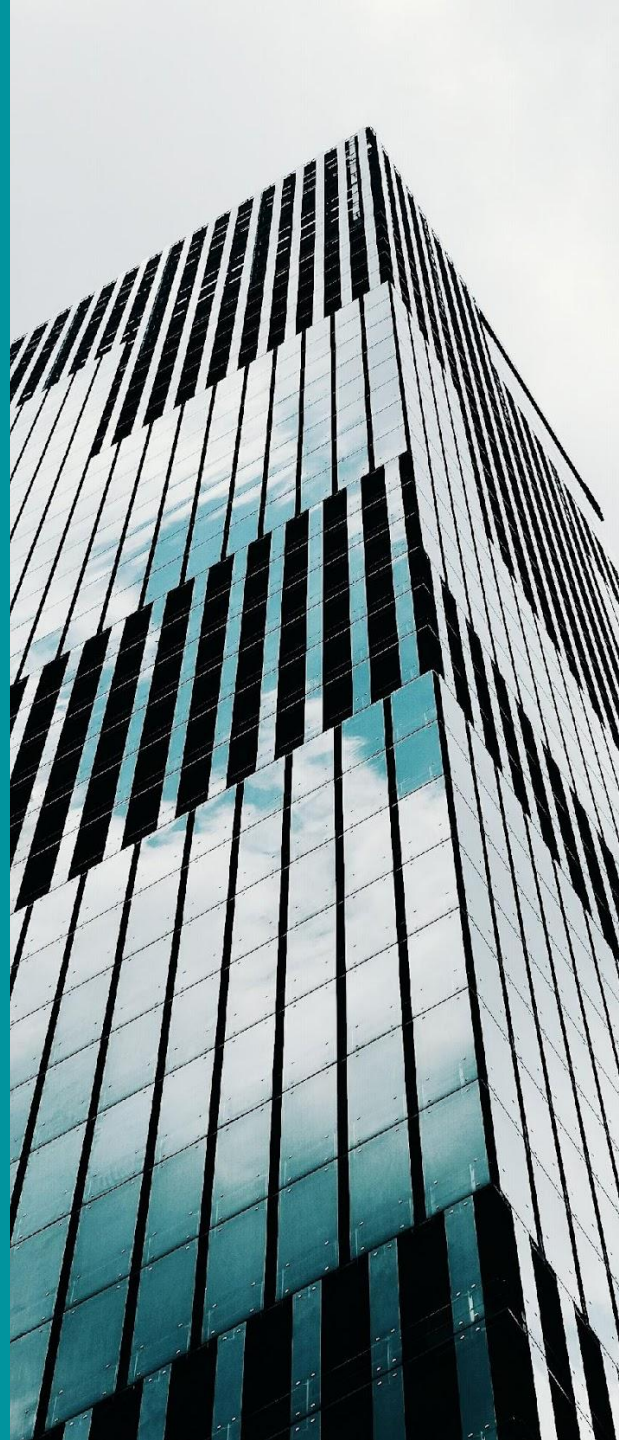
```python
q = '''
    SELECT
        city,
        count(distinct user_id) as total_reseller
    FROM dbt_prod.gross_reseller
    WHERE city is not null
    GROUP BY 1
    ORDER BY 2 DESC
    LIMIT 10
'''

df_user = queryR(q)
df_user
```

```
Connecting to the database . . .
Debug1
Connected to Redshift
```

	city	total_reseller
0	Kota Bandung	
1	Kabupaten Bogor	
2	Kabupaten Bandung	
3	Kota Jakarta Timur	
4	Kabupaten Sleman	





Rakamin Academy

Danke Schön!

Muhammad Ammar Fauzan
https://www.linkedin.com/in/muhammad-ammar-fauzan-748883117/