

Nama : Salmaa Khoirun Nisaa'

NIM : L200180019

Kelas : A

## Modul 1

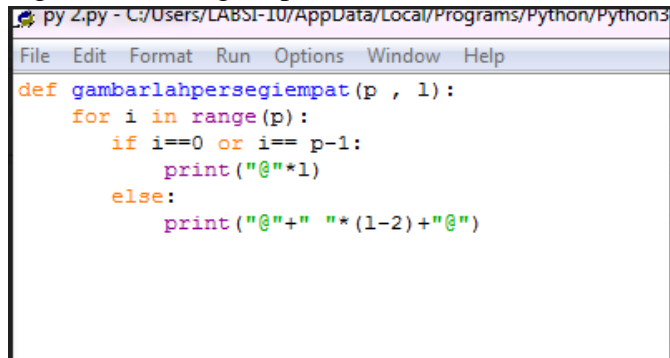
### Tinjauan Ulang Python

#### 1. Fungsi cetakSiku()

```
def cetaksiku(x):  
    for i in range(x+1):  
        print("*" * i)  
  
cetaksiku(5)
```

```
=====  
  
*  
**  
***  
****  
*****
```

#### 2. Fungsi Gambar segiempat



```
>>> gambarlahpersegiempat(4,5)  
@@@@@  
@   @  
@   @  
@@@@@  
>>> |
```

#### 3. Huruf Vokal dan Huruf Konsonan

```

def jumlahHurufVokal(kata):
    vokal = ['a', 'i', 'u', 'e', 'o']
    kata2 = kata.lower()
    list_kata = []
    list_kata.append(len(kata2))
    count = 0
    for i in kata2:
        if i in vokal:
            count += 1
    list_kata.append(count)
    return list_kata

def jumlahHurufKonsonan(kata):
    konsonan = ['b', 'c', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'm', 'n', 'p', 'q', 'r', 's', 't', 'v', 'w', 'x', 'y', 'z']
    kata2 = kata.lower()
    list_kata = []
    list_kata.append(len(kata2))
    count = 0
    for i in kata2:
        if i in konsonan:
            count += 1
    list_kata.append(count)
    return list_kata

```

Activate Windows

```

>>> k=jumlahHurufVokal('Surakarta')
>>> k
[9, 4]
>>> k=jumlahHurufKonsonan('Surakarta')
>>> k
[9, 5]
>>>

```

#### 4. Menghitung rerata

```

import statistics

def rerata(x):
    hasil = float(sum(x)) / max(len(x), 1)
    return hasil

def hitungVariance(x):
    hasil = statistics.variance(x)
    return hasil

def hitungStDev(x):
    hasil = statistics.stdev(x)
    return hasil

```

===== RESTART: C:/U:

```

>>> rerata([1,2,3,4,5])
3.0
>>> g = [3,4,5,4,3,4,5,2,2,10,11,23]
>>> rerata(g)
6.333333333333333
>>>

```

#### 5. Bilangan Prima

```

from math import sqrt as sq

def apakahPrima(n):
    n = int(n)
    assert n >= 0
    primaKecil = [2,3,5,7,11]
    bukanPrKecil = [0,1,4,6,8,9,10]
    if n in primaKecil:
        return True
    elif n in bukanPrKecil:
        return False
    else:
        for i in range(2, int(sq(n)) + 1):
            if n % i == 0:
                return False
        return True

>>> apakahPrima(17)
True
>>> apakahPrima(97)
True
>>> apakahPrima(123)
False
>>>
=====

```

## 6. Cetak bilangan prima

```

for i in range(2, 1001):
    for j in range(2, i//2):
        if (i % j) == 0:
            break
    else:
        print(i)

```

File	Edit	Shell	Debug
715			
727			
733			
739			
743			
751			
757			
761			
769			
773			
787			
797			
809			
811			
821			
823			
827			
829			
839			
853			
857			
859			
863			
877			
881			
883			
887			
907			
911			
919			
929			
937			
941			
947			
953			
967			
971			
977			
983			
991			
997			
>>>			

## 7. Faktor prima

```

from math import sqrt as sq

def faktorPrima(n):
    assert n >= 0
    hasil = []
    while n % 2 == 0:
        hasil.append(2)
        n = n / 2
    for i in range(3, int(sq(n))+1, 2):
        while n % i == 0:
            hasil.append(i)
            n = n / i
    if n > 2:
        hasil.append(int(n))

    return hasil

```

```

=====
>>> faktorPrima(10)
[2, 5]
>>> faktorPrima(120)
[2, 2, 2, 3, 5]
>>> faktorPrima(19)
[19]
>>>

```

#### 8. Fungsi terkandung string

```

def apakahTerkandung(a, b):
    if a in b:
        return True
    else:
        return False

```

```

>>> h= 'do'
>>> k= 'Indonesia Tanah Air'
>>> k= 'Indonesia Tanah Air Beta'
>>> apakahTerkandung(h,k)
True
>>> apakahTerkandung('pusaka', k)
False
>>>

```

#### 9. Program cetak angka 1-100

```

for i in range(1, 101):
    if (i % 3 == 0 and i % 5 == 0):
        print("Python UMS")
    elif i % 3 == 0:
        print("Python")
    elif i % 5 == 0:
        print("UMS")
    else:
        print(i)

```

```

61
62
Python
64
UMS
Python
67
68
Python
UMS
71
Python
73
74
Python UMS
76
77
Python
79
UMS
Python
82
83
Python
UMS
86
Python
88
89
Python UMS
91
92
Python
94
UMS
Python
97
98
Python
UMS
>>>

```

```

from math import sqrt as akar

def selesaikanABC(a, b, c):
    a = float(a)
    b = float(b)
    c = float(c)
    D = b**2 - 4*a*c
    if D < 0:
        print('Determinannya negatif. Persamaan tidak mempunyai akar real')
        quit()
    x1 = (-b + akar(D)) / (2*a)
    x2 = (-b - akar(D)) / (2*a)
    hasil = (x1, x2)
    return hasil

===== RESTART: C:/Users/Salmaa Khoirun N.
>>> selesaikanABC(1,2,3)
Determinannya negatif. Persamaan tidak mempunyai akar real
>>>

```

11. Tahun kabisat

```

def apakahKabisat(tahun):
    if tahun % 4 == 0 or tahun % 400 == 0 and tahun % 100 != 0:
        return True
    else:
        return False

```

Activate Windows  
Go to Settings to activate Windows.

```

>>> apakahKabisat(1896)
True
>>> apakahKabisat(1897)
False
>>>

```

12. Tebak angka

```

from random import randint

quiz = randint(1, 100)

print("Saya menyimpan angka bulat antara 1 sampai 100. coba tebak")

jawab = 0
count = 1
while jawab != quiz:
    jawab = input('Masukkan tebakan ke-{}:>'.format(count))
    jawab = int(jawab)
    if jawab == quiz:
        print('Ya. Anda benar')
    elif jawab < quiz:
        print('Itu terlalu kecil. Coba lagi')
    else:
        print('Itu terlalu besar. Coba lagi')
    count += 1

```

```

===== RESTART: C:/Users/Salmaa Khoirun
Saya menyimpan angka bulat antara 1 sampai 100. coba tebak
Masukkan tebakan ke-1:>50
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-2:>75
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-3:>58
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-4:>56
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-5:>59
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-6:>65
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-7:>70
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-8:>67
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-9:>66
Ya. Anda benar
>>>

```

### 13. Fungsi katakan()



```

def katakan(angka):
    di_bawah20 = ['Nol', 'Satu', 'Dua', 'Tiga', 'Empat', 'Lima', 'Enam', 'Tujuh', 'Delapan',
                  'Sembilan', 'Sepuluh', 'Sebelas', 'Dua belas', 'Tiga belas', 'Empat belas', 'Lima belas',
                  'Enam belas', 'Tujuh belas', 'Delapan belas', 'Sembilan belas']
    puluhan = ['Dua puluh', 'Tiga puluh', 'Empat puluh', 'Lima puluh', 'Enam puluh', 'Tujuh puluh',
               'Delapan puluh', 'Sembilan puluh']
    di_atas100 = {
        100 : 'ratus',
        1000 : 'ribu',
        1000000 : 'juta',
    }

    assert angka >= 0
    assert angka < 10000000000

    if angka < 20:
        return di_bawah20[angka]

    if angka < 100:
        return puluhan[(int)(angka/10)-2] + (' if angka % 10==0 else ' ' + di_bawah20[angka % 10])

    pivot = max([key for key in di_atas100.keys() if key <= angka])

    # Recursion
    hasil = katakan((int)(angka/pivot)) + ' ' + di_atas100[pivot] + (' if angka % pivot==0 else ' ' + katakan(angka % pivot))

    if 'Satu ratus' in hasil:
        hasil = hasil.replace('Satu ', 'Se')

    return hasil.capitalize()

===== RESTART: C:/Users/Salmaa Khoirun Nisaa/AppD
>>> katakan(3125750)
'Tiga juta seratus dua puluh lima ribu tujuh ratus lima puluh'
>>> katakan(1567865)
'Satu juta lima ratus enam puluh tujuh ribu delapan ratus enam puluh lima'
>>>

```

#### 14. Fungsi formatRupiah()

```

def formatRupiah(angka):
    assert angka >= 0
    konversi = 'Rp {0:,}'.format(angka)
    konversi = konversi.replace(',', '.')
    return konversi

>>> formatRupiah(1500)
'Rp 1.500'
>>> formatRupiah(2560000)
'Rp 2.560.000'
>>>

```