



Pemrograman Berbasis Fungsi - RA

TA Genap 2021/2022

Lecturer: Riksa Meidy Karim , S.Kom. , M.Si., M.Sc.

NAMA : KHOIRUNNISA

NIM : 120450029

Tugas Mingguan

>> Pertemuan 1 >>

1.a. jelaskan menggunakan tabel mengenai kelebihan dan kekurangan paradigma prosedural dibandingkan dengan objek oriented?

| | kelebihan | Kekurangan |
|----------------|---|---|
| Objek oriented | <ol style="list-style-type: none">1. Menggunakan class 1x2. Menambahkan fitur class tanpa mengedit class kecil3. Kemudahan pengolahan | <ol style="list-style-type: none">1. Memiliki ruang penyimpanan2. Mudah mengembalikan kode menjadi kode sumber (algoritma yang digunakan pun sulit untuk disembunyikan & mudah dibajak)3. Properti software tidak terlihat dalam satu unit fungsional |

| | | |
|------------|---|--|
| prosedural | <ol style="list-style-type: none"> 1. Penulisan program yang sederhana & terstruktur 2. Memiliki dokumentasi yang baik 3. Memiliki ruang penyimpanan sedikit | <ol style="list-style-type: none"> 1. Program cukup sulit untuk proses perawatan 2. Butuh usaha yang keras untuk menerjemahkan Business Models dalam programming models 3. Fungsi yang tersedia susah untuk diubah tanpa harus mempengaruhi fungsi sistem keseluruhan |
|------------|---|--|

1.b. jelaskan menggunakan tabel mengenai kelebihan dan kekurangan paradigma objek oriented dibandingkan dengan objek functional?

| | Kelebihan | Kekurangan |
|----------------|--|--|
| Objek oriented | <ol style="list-style-type: none"> 1. Memanfaatkan inheritance kode yang berlebihan dihilangkan, kelas yang ada di perpanjang 2. Sistem dapat dengan mudah ditingkatkan dari sistem kecil ke besar 3. Penggunaan kembali kode lebih mudah dari pada bahasa pemrograman konvensional | <ol style="list-style-type: none"> 1. Ukuran besar program menyebabkan kecepatan eksekusi menjadi lambat 2. Memori oop > memory prosedur 3. Mengimplementasikan program cukup rumit & memberatkan komputer |
| Functional | <ol style="list-style-type: none"> 1. Pure functions lebih mudah dipahami 2. Kode lebih mudah dibaca dan dimengerti 3. Pengujian dan debugging lebih mudah | <ol style="list-style-type: none"> 1. Penulisan pure function dapat mengurangi keterbatasan kode 2. Penulisan program dengan gaya rekursif dari pada menggunakan loop dapat sedikit mengintimidasi 3. Menggabungkan pure functions dengan rest of applications dan 1 atau 0 adalah tugas yang sulit |

1.c. jelaskan menggunakan tabel mengenai kelebihan dan kekurangan paradigma prosedural dibandingkan dengan functional?

| | Kelebihan | Kekurangan |
|------------|---|--|
| prosedural | <ol style="list-style-type: none">1. Programnya sederhana& terstruktur2. Implementasi mudah untuk compiler & intepreter3. Memori yang digunakan lebih sedikit4. Lebih mudah ditelusuri | <ol style="list-style-type: none">1. Perawatan program lebih sulit2. Fungsi susah untuk diubah tanpa harus mempengaruhi fungsi sistem keseluruhan3. Tidak memiliki akses modifiers seperti public, protected, dan private |
| Functional | <ol style="list-style-type: none">1. Kode lebih mudah dibaca&dimengerti2. Pengujian & debugging lebih mudah3. Purefunction lebih mudah dipahami | <ol style="list-style-type: none">1. Terkadang menulis purefunction dapat mengurangi terbacanya kode2. Menulis program rekursif dari pada menggunakan loop dapat sedikit mengintimidasi3. Nilai-nilai tidak dapat menurunkan kinerja |

2.a. apa yang di maksud dengan pure functional programming language?

Adalah paradigma pemrograman yang ada sebelumnya muncul objek oriented. Dengan memperlakukan fungsi dan data sebagai immutable (tidak berubah) karena function tidak boleh mengubah data asli atau status program.

2.b. dari bahasa pemrograman di slide 14, manakah yang termasuk pure functional program language?

1. java; 2. Matlab; 3. PHP; 4. C++; 5. Rust; 6. Closure

>> Pertemuan 2 >>

1. Pilih 3 built in function dalam dokumentasi berikut:

Jawab :

`abs()`, `all()`, `any()`, `ascii()`, `bin()`, `bool()`, `bytearray()`, `bytes()`

2. Deskripsikan apa yang dikomputasikan oleh fungsi tersebut!

Jawab :

`abs()` = mengembalikan nilai absolut suatu bilangan

`all()` = mengembalikan True jika semua anggota data iterable bernilai True

`any()` = menguji apakah satu atau lebih anggota iterable bernilai True

`ascii()` = mengembalikan string yang berisi karakter yang bisa dicetak

`bool()` = mengubah suatu nilai menjadi boolean

`bin()` = mengubah integer menjadi string biner

`bytearray()` = mengembalikan byte array

`bytes()` = mengembalikan objek byte immutable

3. Implementasikan 3 built function tersebut!

Jawab :

```
l = [1, 2, 3]
```

```
Print (all(l))
```

```
l = [1, False, 3]
```

```
print(all(l))
```

```
l = []
```

```
print(all(l))
```

```
test = []
```

```
print(test, 'adalah', bool(test))
```

```
test = [0]
```

```
print(test, 'adalah', bool(test))
```

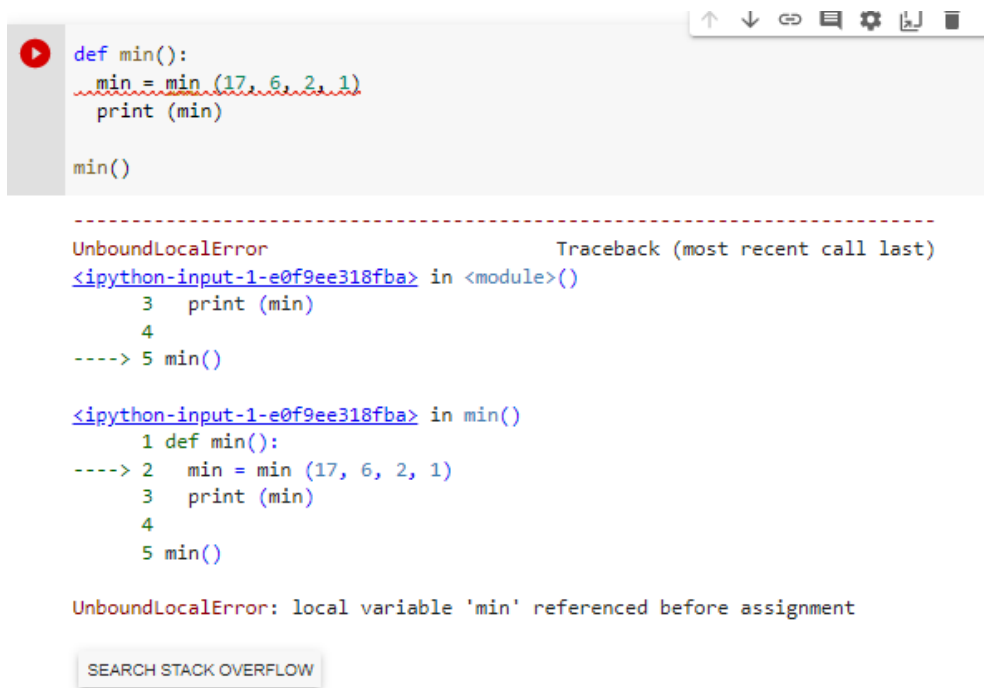
```
angka = 2
```

```
print("Nilai biner dari 2 adalah", bin(angka))
```

4. Apa yang terjadi saat kita memberikan nama fungsi yang kita define sama dengan fungsi built in function? Example (fungsi max()). Sertakan ss program!

Jawab :

Jika user defined sama dengan built in function maka program tidak berjalan/error



```
def min():
    min = min(17, 6, 2, 1)
    print(min)

min()
```

UnboundLocalError Traceback (most recent call last)

<ipython-input-1-e0f9ee318fba> in <module>()

3 print (min)

4

----> 5 min()

<ipython-input-1-e0f9ee318fba> in min()

1 def min():

----> 2 min = min(17, 6, 2, 1)

3 print (min)

4

5 min()

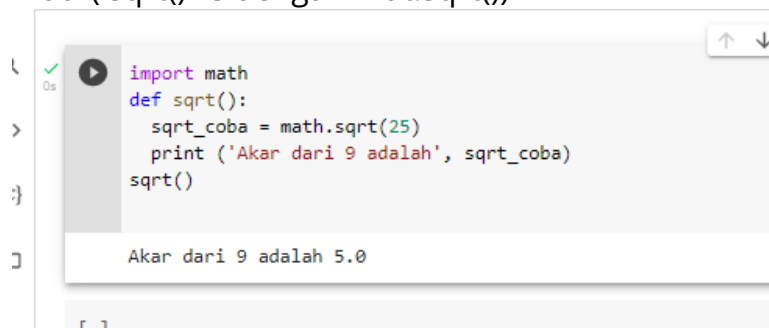
UnboundLocalError: local variable 'min' referenced before assignment

SEARCH STACK OVERFLOW

5. Apa yang terjadi saat kita memberikan nama fungsi yang kita define sama dengan fungsi dalam module? Exx (fungsi sqrt() vs math sqrt()). Sertakan ss program!

Jawab :

Syntax tetap berjalan karena user defined function berbeda dengan module math(sqrt() vs dengan mat.sqrt())



```
import math
def sqrt():
    sqrt_coba = math.sqrt(25)
    print ('Akar dari 9 adalah', sqrt_coba)
sqrt()
```

Akar dari 9 adalah 5.0

```

1 def f(x):
2     return x**3-5*x-9
3 def bisection(x0,x1,e):
4     step = 1
5     print('\n\n*** BISECTION METHOD IMPLEMENTATION ***')
6     condition = True
7     while condition:
8         x2 = (x0 + x1)/2
9         print('Iteration-%d, x2 = %0.6f and f(x2) = %0.6f' % (step, x2, f(x2)))
10
11         if f(x0) * f(x2) < 0:
12             x1 = x2
13         else:
14             x0 = x2
15
16         step = step + 1
17         condition = abs(f(x2)) > e
18
19     print('\nRequired Root is : %0.8f' % x2)
20 x0 = input('First Guess: ')
21 x1 = input('Second Guess: ')
22 e = input('Tolerable Error: ')
23 x0 = float(x0)
24 x1 = float(x1)
25 e = float(e)
26 if f(x0) * f(x1) > 0.0:
27     print('Given guess values do not bracket the root.')
28     print('Try Again with different guess values.')
29 else:
30     bisection(x0,x1,e)

```

6.

7. Dari code diatas , apa saja fungsi yang termasuk user defined definition?

Jawab :

F(x), bisection(), x0,x1, e, x0,x1

8. Apa saja fungsi yang termasuk built in function?

Jawab :

float(), print(), input(),

9. Identifikasikan pada line berapa saja yang terdapat formal parameter dan actual parameter! Tuliskan juga parameter dan fungsi nya!

Jawab :

Formal parameter : line 1 > (x)

line 3 > (x0,x1,e) in def bisection

Actual parameter : line 2 > x**3-5*x-9

line 8 > x2 = (x0+x1)/2

line 17 > condition = abs (f(x2)) > 2

line 20 > x0

line 21 > x1

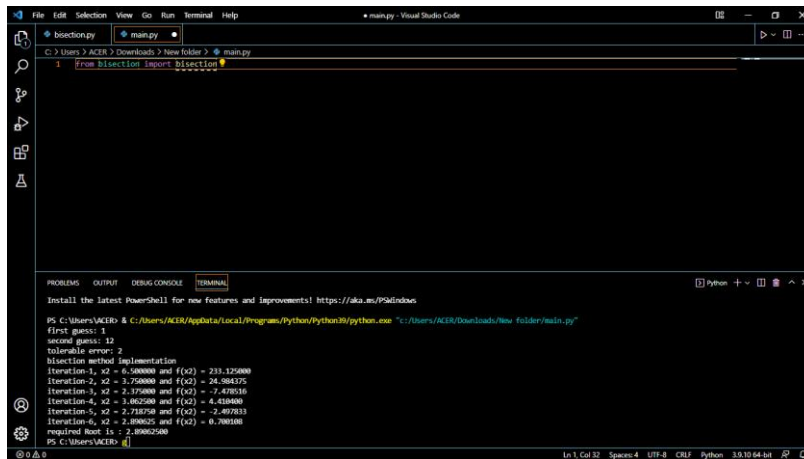
line 22 : e

10. Tulis kembali code tersebut, kemudian screenshot output program nya!

```
def f(x):  
    return x**3-5*x-9  
def bisection(x0, x1, e):  
    step = 1  
    print ("bisection method implementation")  
    condition = True  
    while condition :  
        x2 = (x0 + x1)/2  
        print ("iteration-%d, x2 = %0.6f and f(x2) = %0.6f" %(step, x2, f(x2)))  
  
        if f(x0) * f(x2) < 0:  
            x1 = x2  
        else :  
            x0 = x2  
        step = step + 1  
        condition = abs(f(x2)) > e  
  
    print ("required Root is : %0.8f" % x2)  
  
x0 = input ("first guess: ")  
x1 = input ("second guess: ")  
e = input ("tolerable error: ")  
x0 = float(x0)  
x1 = float(x1)  
e = float(e)  
if f(x0) * f(x1) > 0.0:  
    print (" given guess value don't bracket the root")  
    print ("try again with different guess values")  
else:  
    bisection(x0, x1, e)
```

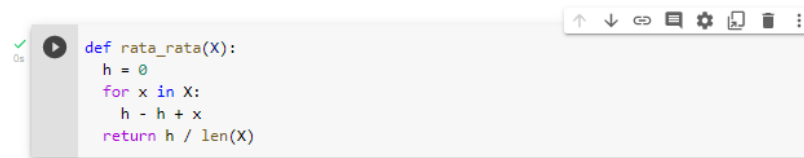
```
first guess: 2  
second guess: 20  
tolerable error: 0.1  
bisection method implementation  
iteration-1, x2 = 11.000000 and f(x2) = 1267.000000  
iteration-2, x2 = 6.500000 and f(x2) = 233.125000  
iteration-3, x2 = 4.250000 and f(x2) = 46.515625  
iteration-4, x2 = 3.125000 and f(x2) = 5.892578  
iteration-5, x2 = 2.562500 and f(x2) = -4.986084  
iteration-6, x2 = 2.843750 and f(x2) = -0.221588  
iteration-7, x2 = 2.984375 and f(x2) = 2.658443  
iteration-8, x2 = 2.914062 and f(x2) = 1.175208  
iteration-9, x2 = 2.878906 and f(x2) = 0.466135  
iteration-10, x2 = 2.861328 and f(x2) = 0.119621  
iteration-11, x2 = 2.852539 and f(x2) = -0.051645  
required Root is : 2.85253906
```

11. Buatlah modul python bersama bisection.py dari code tersebut, kemudian buat file bernama main.py. Panggil fungsi bisection dari main.py! Screenshot output programnya!



```
from bisection import bisection
bisection()

PS C:\Users\ACER> & C:\Users\ACER\AppData\Local\Programs\Python\Python39\python.exe "C:\Users\ACER\Downloads\New folder\main.py"
first guess: 1
second guess: 12
tolerable error: 2
bisection method implementation
Iteration-1, x2 = 6.500000 and f(x2) = 233.125000
Iteration-2, x2 = 3.750000 and f(x2) = 24.584375
Iteration-3, x2 = 2.375000 and f(x2) = -2.458134
Iteration-4, x2 = 3.062500 and f(x2) = 4.418600
Iteration-5, x2 = 2.718750 and f(x2) = -2.407833
Iteration-6, x2 = 2.890625 and f(x2) = 0.780588
required root is : 2.89602548
PS C:\Users\ACER>
```



```
def rata_rata(X):
    h = 0
    for x in X:
        h = h + x
    return h / len(X)
```

terdapat sebuah fungsi rata_rata(X) yang menerima input parameter berupa list. Fungsi tersebut mengouputkan rata-rata dari List tersebut.

12.

Jelaskan apakah fungsi rata_rata(X) diatas dapat diubah menjadi fungsi lambda?

Jawab :

Fungsi diatas dapat diubah menjadi fungsi lambda

13. **Modularisasikan fungsi tersebut (breakdown menjadi beberapa fungsi)!**

Jawab :

Def rata_rata(X) : ialah langkah pertama membuat user defined function

h = 0 : mendefinisikan h bernilai 0

for x in X : mendefinisikan x ke X

h = h + x : memasukkan h dengan fungsi h + x

Return h/ len(X) : perulangan kembali ke h atau menjalankan len(X)

-
14. Ubahlah fungsi rata_rata(X) tersebut menjadi fungsi lambda!
(tidak boleh menggunakan fungsi built in sum ())

```
1 def func(x):  
2     return lambda : x * x  
3  
4 a = func (4)  
5 b = func (5)  
6 print(a())  
7 print (b())
```