



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231058
Nama Lengkap	Michael Chandra Mahanaim
Minggu ke / Materi	04 / Modular Programming

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA

BAGIAN 1: MATERI MINGGU INI (40%)

MATERI 1: Fungsi

Fungsi adalah cara mudah untuk mempacketkan suatu kode agar bisa digunakan / dipanggil berkali-kali tanpa harus menetik ulang isi kode. Fungsi seperti ini sangat penting digunakan, terutama dalam game seperti RPG, dimana kalkulasi damage harus dihitung berkali-kali. Daripada menetik kode berkali-kali, lebih efisien jika menggunakan Fungsi. Contohnya adalah sebagai berikut:

```
import random

def damage(hp):
    hit = random.randint(1,10)
    hp -= hit
    return hp

health = 100
while health > 0:
    act = input("Action: ")
    if act == "attack":
        health = damage(health)
    else:
        print("Invalid")
    print(health)
```

Disamping adalah contoh kalkulasi damage di sebuah RPG. Fungsi memiliki nama "damage" dan variabel kondisi meminta "health" dari player dan meminta aksi yang akan dipilih oleh user. Untuk menyederhanakan, hanya ada aksi "attack".

Setelah menerima aksi, maka fungsi akan dipanggil dan kalkulasi akan terjadi, variabel "hit" akan membuat angka random dari 1-10 dan mengurangi "hp" dengan angka tersebut sehingga dapat mengeluarkan return hasil kalkulasi dan di print.

Fungsi memiliki banyak kegunaan lain selain untuk ini, dapat juga dilihat dalam contoh selanjutnya.

```
def calculator(n1, n2, op):
    if op == "+":
        r = n1+n2
    elif op == "-":
        r = n1-n2
    elif op == "*":
        r = n1*n2
    elif op == "/":
        r = n1/n2
    else:
        return ("Invalid operator")
    return r
```

Contoh di samping adalah penggunaan fungsi dalam bentuk kalkulator. Variabel yang diperlukan hanyalah 2 angka dan sebuah operator. Setelah itu maka kode akan berjalan dengan variabel dan kondisi yang sudah dibuat. Setelah selesai maka r akan di return sebagai hasil dari kalkulasi.

Fungsi sangatlah berguna jika ingin mengompress sebuah kode agar tidak terlihat berserakan dan terlihat simpel.

MATERI 2: Return Value

Pada sebuah fungsi, posisi sangatlah penting. Dalam praktis, fungsi sebaiknya selalu ditaruh pada awal atau bagian teratas sebuah kode. Karena eksekusi sebuah kode berjalan dari atas ke bawah, maka semua fungsi yang dipanggil sebelum di def (define / definisikan) tidak akan terbaca dan menyebabkan sintaks error. Seperti contoh di bawah ini:

```
42  r = subt(10)
43
44  def subt(x):
45      x -= 10
46      return x
47
48  print(r)
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE

⊗ PS C:\Users\micha\Documents\College\2S
aspralpro4/SrcCd.py

○ Traceback (most recent call last):
File "c:\Users\micha\Documents\College\2S\aspralpro4/SrcCd.py", line 42, in <module>
r = subt(10)
^^^^
NameError: name 'subt' is not defined

Gambar sintaks error fungsi misplace

Gambar diatas langsung menyatakan bahwa sudah ada problem dan jika kode di jalankan maka sintaks error yang keluar terjadi karena fungsi "subt" tidak di defined. Selain posisi, jangan lupa untuk menambahkan return value supaya kode dapat berfungsi dengan lancar.

Untuk memperjelas return value, ada juga sebuah void function dimana tidak menggunakan return tetapi hanya menjalankan kode untuk memprint. Seperti contoh fungsi di bawah ini:

```
• 50 ~ def prt2(message):
51     print(message)
52     print(message)
53
54 prt2("Good morning")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
• PS C:\Users\micha\Documents\College\2SMT\TPA
aspralpro4\SrcCd.py
Good morning
Good morning
```

Contoh void function dengan print

Fungsi di atas langsung melaksanakan sintaks print dan mengeluarkan output sesuai dengan perintah dari fungsi tersebut. Di sisi lainnya ada juga return value. Return value dapat berupa string, float, integer, dan boolean. Return value sangat berguna untuk mengolah data, misalnya ingin mengolah hasil perpangkatan dan mengeluarkan hasil, maka kode tersebut akan terlihat seperti di bawah ini jika menggunakan fungsi:

```
def powered(x, y):
    r = x**y
    return r

n1 = input("Number: ")
n2 = input("Power: ")

result = powered(n1, n2)
print(result)
```

Gambar kode singkat perpangkatan dengan fungsi

Kode di atas memanggil fungsi "powered" dan mengolah 2 input user (n1 dan n2) sehingga menghasilkan hasil perpangkatan dari $n1^{n2}$, dan setelah hasil ditemukan, maka di return supaya dapat menjadi value dari variabel "result". Setelah itu tinggal di print dan sudah hasil akan ditampilkan.

MATERI 3: Optional Argument dan Named Argument

Named argument adalah untuk memberi nilai default kepada sebuah fungsi agar dapat berjalan langsung tanpa memberi input / jika input kosong. Contohnya ada di bawah ini:

```
def fprice(amnt=1, cost=1000):  
    final_price = amnt * cost  
    return final_price  
  
fp = fprice()  
print(fp)
```

```
PS C:\Users\micha\De  
aspralpro4\SrcCd.py  
1000
```

Gambar kode dan output singkat contoh named argument

Kode diatas dapat berjalan dengan lancar tanpa memberi variabel untuk fungsi fprice() karena sudah diberi named argument / default value untuk kedua variabel kondisi fprice(). Kode ini juga dapat di expand jika ingin diubah, misalkan amnt ingin diubah menjadi 5 maka hanya perlu menambahkan tambahan pada kode fp = fprice(5), dengan ini variabel "amnt" sekarang bernilai 5 dan output juga akan mengeluarkan hasil yang sama, seperti di bawah ini:

```
def fprice(amnt=1, cost=1000):  
    final_price = amnt * cost  
    return final_price  
  
fp = fprice(5)  
print(fp)
```

```
PS C:\Users\micha\Documents\College\2SMT\TPAP1\tugas  
aspralpro4/SrcCd.py  
5000
```

Gambar kode dan output dari perubahan value variabel amnt

Hasil yang keluar akan sesuai dengan input variabel terbaru. Default value hanya akan digunakan jika saat ingin mengkalkulasi ada variabel yang tidak memiliki value. Maka input dari user juga masih dapat digunakan dengan mudah.

MATERI 4: Anonymous Function (Lambda)

Lambda, secara singkat adalah versi mini dari function. Dapat di call dan dapat menggunakan if else, logic, dan aritmatika. Perbedaannya hanyalah jika fungsi menggunakan cara kode biasa yang turun ke bawah, lambda menggunakan ternary yang membuat sebuah fungsi sederhana dalam 1 baris saja. Contoh penggunaan lambda adalah:

```
38 addten = lambda n: n+10  
39  
40 print(addten(5))  
  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
  
PS C:\Users\micha\Documents\College\2SMT\TPAP1\tugas  
aspralpro4/SrcCd.py  
15
```

Gambar penggunaan lambda mirip fungsi

Pada kode di atas, lambda tidak menggunakan def tetapi menggunakan variabel "addten" = lambda, setelah kata lambda diberi variabel argumen seperti untuk fungsi calculator di atas meminta (n1, n2, op), fungsi lambda ini meminta n. Setelah ":" maka dimasukkan operasi yang akan dijalankan, pada kode ini hanya diminta untuk menjumlahkan n + 10. Untuk memanggil / call fungsi lambda ini hanya perlu mengetik nama variabel (addten) dan disampingnya diberi tanda kurung beserta variabel yang akan jadi argumen (n). Di contoh diatas dimasukkan 5, maka 5 + 10 adalah 15 dengan begitu output yang keluar adalah 15.

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

A. Source Code

```
9
10 def cek_angka(n1, n2, n3):
11     if (n1 != n2 != n3 != n1) and (n1+n2 == n3 or n2+n3 == n1 or n1+n3 == n2):
12         return True
13     else:
14         return False
15
16 num1 = input("Input 1st number: ")
17 num2 = input("Input 2nd number: ")
18 num3 = input("Input 3rd number: ")
19
20 try:
21     num1 = int(num1)
22     num2 = int(num2)
23     num3 = int(num3)
24 except:
25     print("Please use numbers, not words")
26     exit()
27
28 result = cek_angka(num1, num2, num3)
29 print(result)
```

B. Output Result

Input	Output
Input 1st number: 20	Input 1st number: 20
Input 2nd number: 30	Input 2nd number: 20
Input 3rd number: 10	Input 3rd number: 40
True	False

C. Explanation

Kode ini menggunakan fungsi serta input user, simple logic(and dan or), dan try except. Fungsi digunakan untuk menyimpan logic penentu output sesuai dengan kondisi. Input user digunakan agar lebih fleksibel, dan try except digunakan untuk mengatasi kesalahan input user.

SOAL 2

A. Source Code

```
Welcome | Question1.py U | Question2.py U x | Question3.py U
Question2.py > ...
12
13 def cek_digit_belakang(n1, n2, n3):
14     tn1 = n1 % 10
15     tn2 = n2 % 10
16     tn3 = n3 % 10
17     if (tn1 == tn2 or tn2 == tn3 or tn1 == tn3):
18         return True
19     else:
20         return False
21
22 num1 = input("Insert 1st number: ")
23 num2 = input("Insert 2nd number: ")
24 num3 = input("Insert 3rd number: ")
25
26 try:
27     num1 = int(num1)
28     num2 = int(num2)
29     num3 = int(num3)
30 except:
31     print("Please use numbers, not words")
32     exit()
33
34 result = cek_digit_belakang(num1, num2, num3)
35 print(result)
```


B. Output Result

<pre>PS C:\Users\micha\Documents\aspralpro4\Question2.py Insert 1st number: 1802 Insert 2nd number: 18592 Insert 3rd number: 177 True</pre>	<pre>PS C:\Users\micha\Documents\aspralpro4\Question2.py Insert 1st number: 20 Insert 2nd number: 1764 Insert 3rd number: 192436 False</pre>
---	--

C. Explanation

Kode ini sama seperti soal 1, menggunakan fungsi, simple logic, modulo, user input, dan try except. Fungsi digunakan untuk memproses input dengan, modulo digunakan untuk mendapatkan angka paling kanan, dan logic digunakan untuk menentukan hasil output sesuai dengan syarat yang diinginkan soal, terakhir try except digunakan untuk mengatasi kesalahan input user.

SOAL 3

A. Source Code

```
Welcome × Question1.py U Question2.py U Question3.py U ×
Question3.py > ...
11 fhr = lambda c: (9/5)*c +32
12 rem = lambda c: 0.8*c
13
14 celc = input("Input temp to convert: ")
15
16 try:
17     celc = int(celc)
18 except:
19     print("Please us numbers, not words")
20     exit()
21
22 print(f"Temprature in Fahrenheit: {fhr(celc)}")
23 print(f"Temprature in Reamur: {rem(celc)}")
```

B. Output Result

Input	Fahrenheit	Reamur
0	32.0	0.0
100	212.0	80.0

C. Explanation

Kode ini, agak berbeda dari soal sebelumnya. Kode ini menggunakan fungsi dalam bentuk lambda. Lambda adalah fungsi def() dalam bentuk kecil dan singkat. Kode ini juga menggunakan user input dan try except untuk mengatasi kesalahan input. Penggunaan lambda relatif mudah, hanya perlu menulis lambda <variabel>: <isi/operasi/ekspresi>.