



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	<71230972>
Nama Lengkap	<Oktavian Christ Putranto>
Minggu ke / Materi	04/Modular Programing

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  
YOGYAKARTA  
2024

## BAGIAN 1: MATERI MINGGU INI (40%)

### Fungsi, Argument dan Parameter

Fungsi yang bias akita pakai di python seperti fungsi `input()` yang berfungsi untuk membaca inputan pengguna dan `print()` yang berfungsi untuk menampilkan output tulisan, kedua fungsi yang sering kita gunakan ini termasuk dalam salah satu fungsi bawaan yang ada dalam python fungsi fungsi lain yang merupakan fungsi bawaan antara lain ialah :

Built-in Functions			
<b>A</b> <a href="#"><u>abs()</u></a> <a href="#"><u>aiter()</u></a> <a href="#"><u>all()</u></a> <a href="#"><u>anext()</u></a> <a href="#"><u>any()</u></a> <a href="#"><u>ascii()</u></a>	<b>E</b> <a href="#"><u>enumerate()</u></a> <a href="#"><u>eval()</u></a> <a href="#"><u>exec()</u></a>	<b>L</b> <a href="#"><u>len()</u></a> <a href="#"><u>list()</u></a> <a href="#"><u>locals()</u></a>	<b>R</b> <a href="#"><u>range()</u></a> <a href="#"><u>repr()</u></a> <a href="#"><u>reversed()</u></a> <a href="#"><u>round()</u></a>
<b>B</b> <a href="#"><u>bin()</u></a> <a href="#"><u>bool()</u></a> <a href="#"><u>breakpoint()</u></a> <a href="#"><u>bytearray()</u></a> <a href="#"><u>bytes()</u></a>	<b>F</b> <a href="#"><u>filter()</u></a> <a href="#"><u>float()</u></a> <a href="#"><u>format()</u></a> <a href="#"><u>frozenset()</u></a>	<b>M</b> <a href="#"><u>map()</u></a> <a href="#"><u>max()</u></a> <a href="#"><u>memoryview()</u></a> <a href="#"><u>min()</u></a>	<b>S</b> <a href="#"><u>set()</u></a> <a href="#"><u>setattr()</u></a> <a href="#"><u>slice()</u></a> <a href="#"><u>sorted()</u></a> <a href="#"><u>staticmethod()</u></a> <a href="#"><u>str()</u></a> <a href="#"><u>sum()</u></a> <a href="#"><u>super()</u></a>
<b>C</b> <a href="#"><u>callable()</u></a> <a href="#"><u>chr()</u></a> <a href="#"><u>classmethod()</u></a> <a href="#"><u>compile()</u></a> <a href="#"><u>complex()</u></a>	<b>G</b> <a href="#"><u>getattr()</u></a> <a href="#"><u>globals()</u></a>	<b>N</b> <a href="#"><u>next()</u></a>	<b>T</b> <a href="#"><u>tuple()</u></a> <a href="#"><u>type()</u></a>
<b>D</b> <a href="#"><u>delattr()</u></a> <a href="#"><u>dict()</u></a> <a href="#"><u>dir()</u></a> <a href="#"><u>divmod()</u></a>	<b>H</b> <a href="#"><u>hasattr()</u></a> <a href="#"><u>hash()</u></a> <a href="#"><u>help()</u></a> <a href="#"><u>hex()</u></a>	<b>O</b> <a href="#"><u>object()</u></a> <a href="#"><u>oct()</u></a> <a href="#"><u>open()</u></a> <a href="#"><u>ord()</u></a>	<b>V</b> <a href="#"><u>vars()</u></a>
	<b>I</b> <a href="#"><u>id()</u></a> <a href="#"><u>input()</u></a> <a href="#"><u>int()</u></a> <a href="#"><u>isinstance()</u></a> <a href="#"><u>issubclass()</u></a> <a href="#"><u>iter()</u></a>	<b>P</b> <a href="#"><u>pow()</u></a> <a href="#"><u>print()</u></a> <a href="#"><u>property()</u></a>	<b>Z</b> <a href="#"><u>zip()</u></a>
			<a href="#"><u>--import__()</u></a>

Secara umum fungsi fungsi tersebut memiliki kegunaan khususnya masing-masing dan dapat digunakan berkali kali untuk kebutuhan yang beragam. Dan penggunaan fungsi sudah menjadi hal wajib untuk dipelajari karena Jika anda suatuwaktu akan membuat program yang membutuhkan Langkah-langkah yang banyak dan rumit, maka pengelompokkan kode-kode program menjadi bagian (block) dari suatu program yang besar merupakan pengertian dasar dari modular programing. Kondisi di mana

program anda terdiri dari beberapa bagian modular yang memiliki kegunaan khusus dan dapat digunakan ulang itulah yang disebut sebagai modular programming.

Fungsi sendiri dibedakan menjadi 2 yaitu fungsi bawaan seperti contoh diatas dan juga fungsi yang dibuat oleh programmer contoh :

```
def.py > tambah
1  def tambah(a, b):
2      |      hasil = a + b
3      |      return hasil
```

Penggunaan nya dalam program :

```
def.py > ...
1  def tambah(a, b):
2      |      hasil = a + b
3      |      return hasil
4
5  c = tambah(10, 5)
6  print(c)
```

Output :

```
PS C:\Users\ASUS VivoBook\Downloads\prak_4> & "C:/Users/ASUS VivoBook/Documents/kuliah/msys64/mingw64/bin/python.exe" "c:/U
sers/ASUS VivoBook/Downloads/prak_4/def.py"
15
```

Hal yang harus diperhatikan dalam urutan fungsi yang dibuat:

- Urutan baris define harus ada diatas recall function nya
- Pastikan tidak ada salah ketik dan pastikan huruf kapital yang dipakai dalam function dan recallnya harus sama
- tidak memakai spasi sebagai pemisah sebaiknya pakai(\_)

Contoh contoh kesalahan yang sering dilakukan dalam pembuatan function:

Salah urutan :

```
def.py > tambah
1  c = tambah(10, 5)
2  print(c)
3
4  def tambah(a, b):
5      hasil = a + b
6      return hasil

sers/ASUS VivoBook/Downloads/prak_4/def.py"
Traceback (most recent call last):
  File "c:\Users\ASUS VivoBook\Downloads\prak_4\def.py", line 1, in <module>
    c = tambah(10, 5)
NameError: name 'tambah' is not defined
```

Dalah contoh diatas ketika function di tempatkan di bawah maka yang terjadi ialah python tidak akan bisa mendeteksi function tersebut karena program python merupakan line base sehingga dibaca dari atas ke bawah sehingga function harus di define terlebih dahulu sebelum di gunakan.

Typo huruf kapital:

```
def.py > Tambah
1  def Tambah(a, b):
2      hasil = a + b
3      return hasil
4
5  c = tambah(10, 5)
6  print(c)

PS C:\Users\ASUS VivoBook\Downloads\prak_4> & "C:/Users/ASUS VivoBook/Documents/kuliah/msys64/mingw64/bin/python.exe" "c:/U
sers/ASUS VivoBook/Downloads/prak_4/def.py"
Traceback (most recent call last):
  File "c:\Users\ASUS VivoBook\Downloads\prak_4\def.py", line 5, in <module>
    c = tambah(10, 5)
NameError: name 'tambah' is not defined. Did you mean: 'Tambah'?
```

Karena python sensitive terhadap penggunaan huruf kapital maka ketika ada salah ketik dalam me recall function maka function tidak akan berjalan/tidak terdeteksi

## Optional Argument dan Named Argument

Dalam pembuatan suatu fungsi kita dapat menambahkan parameter parameter di dalamnya dan parameter ini bis akita tambahkan nilai default yang akan menjadi nilai valuenya ketika parameter tidak di isi, contoh parameter :

```
parameter.py > ...  
1  def hitung_belanja(belanja, diskon=0):  
2      bayar = belanja - (belanja * diskon)/100  
3      return bayar  
4  
5  print(hitung_belanja(100000))  
6  print(hitung_belanja(100000, 10))  
7  print(hitung_belanja(100000, 50))
```

```
● PS C:\Users\ASUS V  
sers/ASUS VivoBook  
100000.0  
90000.0  
50000.0
```

Belanja dan diskon ilah parameter yang di tentukan dalam function, pada bagian parameter `diskon=0` memiliki arti bahwa pada parameter ini telah ditentukan default value sebesar 0 yang berarti ketika tidak terdapat value diskon pada saat recall function terjadi maka otomatis value dari parameter diskon akan bernilai 0.

Untuk me return sebuah fungsi selain memakai print seperti contoh diatas kita juga dapat memakai cara lain yaitu dengan menyebutkan nama argumentnya (named argument). Jika anda menggunakan cara tersebut, maka urutan argument tidak harus sama dengan urutan parameter pada fungsi, contoh :

```
argu.py > ...  
1  def cetak(a, b, c):  
2      print("Nilai a: ",a)  
3      print("Nilai b: ",b)  
4      print("Nilai c: ",c)  
5  cetak(b=30, c=40, a=20)
```

```
● PS C:\Users\ASUS V  
sers/ASUS VivoBook  
Nilai a: 20  
Nilai b: 30  
Nilai c: 40
```

## Lambda

Sederhananya lambda ialah bentuk penulisan lain dalam membuat sebuah function contoh :

```
def.py > ...
1  #function dalam bentuk yang biasanya kita gunakan
2  def Tambah(a, b):
3      |      hasil = a + b
4      |      return hasil
5
6  c = Tambah(10, 20)
7  print(c)
8
9  #function dalam bentuk lambda
10 tambah = lambda a, b: a + b
11 print(tambah(10,20))
12
```

```
● PS C:\Users\
sers/ASUS V
30
30
```

Keduanya memiliki output yang sama dan dapat digunakan sebagaimana function pada umumnya dan memiliki kelebihan yaitu hanya memerlukan 1 line penulisan saja sehingga dapat lebih efisien.

Contoh penggunaan lambda dicampur dengan dictionary:

```
tagihan.py > ...
1  tarif={
2      "1":lambda jarak: (jarak-100)*2000 + 100*1500 if jarak>100 else (jarak*1500),
3      "2":lambda jarak: (jarak-100)*3000 + 100*2500 if jarak>100 else (jarak*2500),
4      "3":lambda jarak: (jarak-100)*5000 + 100*4000 if jarak>100 else (jarak*4000),
5      "4":lambda jarak: (jarak-100)*7000 + 100*5000 if jarak>100 else (jarak*5000),
6      "":lambda jarak: (jarak-100)*5000 + 100*4000 if jarak>100 else (jarak*4000)
7  }
8  jarak = int(input("masukan jarak:"))
9  golongan = input("golongan : ")
10 print(tarif [golongan] (jarak))
11
```

```
sers/ASUS VIVOBOOK/Downloads/prak_4/ta  
masukan jarak:100  
golongan : 2  
250000
```

Pada code ini saya menggunakan dictionary berisi lambda yang dapat menentukan golongan harga untuk membayar tagihan listriknya dan untuk golongan yang tidak dicantumkan maka defaultnya akan menjadi golongan 3.

Fungsi tersebut dalam bentuk normal :

```
listrik.py > ...  
1  def tagih(jarak,gol=3):  
2      if gol==1:  
3          bayar = (jarak-100)*2000 + 100*1500 if jarak>100 else (jarak*1500)  
4          return bayar  
5      elif gol==2:  
6          bayar2 = (jarak-100)*3000 + 100*2500 if jarak>100 else (jarak*2500)  
7          return bayar2  
8      elif gol==3:  
9          bayar3 =(jarak-100)*5000 + 100*4000 if jarak>100 else (jarak*4000)  
10         return bayar3  
11     elif gol==4:  
12         bayar4 = (jarak-100)*7000 + 100*5000 if jarak>100 else (jarak*5000)  
13         return bayar4  
14     print(tagih(100,2))
```

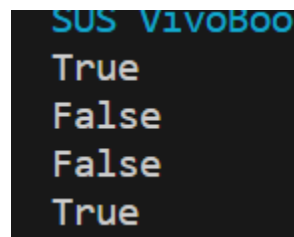
```
sers/ASUS V.  
● 250000
```

## BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

### SOAL 1

```
lat_1.py > ...
1 def cek_angka(a,b,c):
2     if (a!=b!=c!=a)and(a+b == c or b+c == a or c+a == b):
3         return True
4     else:
5         return False
6 print (cek_angka(1,2,3))
7 print (cek_angka(2,2,3))
8 print (cek_angka(5,3,1))
9 print (cek_angka(7,2,5))
10
11 #bentuk function apabila memakai lambda
12 #cek_angka= lambda a,b,c: True if (a!=b!=c!=a and (a+b == c or b+c == a or c+a == b)) else False
13
```



```
SUS VivoBook
True
False
False
True
```

Pada soal pertama kita diminta membuat sebuah fungsi yang dapat mengecek ketiga buah angka apakah memenuhi persyaratan yang ditentukan atau tidak, untuk syarat pertama ialah menentukan apakah ketiganya memiliki nilai yang berbeda dan syarat yang kedua ialah Ada kemungkinan jika diambil dua parameter dan dijumlahkan hasilnya sama dengan parameter lainnya, apabila semua syarat terpenuhi akan mengeluarkan nilai true kalau tidak maka akan mengeluarkan nilai false

Pada code yang saya buat pertama tama saya membuat function dengan nama cek\_angka dengan 3 parameter yaitu a,b,c setelah itu saya membuat conditional if (a!=b!=c!=a) and (a+b == c or b+c == a or c+a == b)

conditional (a!=b!=c!=a) ini menentukan apakah ketiganya memiliki nilai yang berbeda atau tidak kemudian disambung dengan oprator logical and yang berarti kedua conditional harus true agar return true.

Conditional (a+b == c or b+c == a or c+a == b) berguna untuk menentukan apakah ada kemungkinan jika diambil dua parameter dan dijumlahkan hasilnya sama dengan parameter lainnya

Jika salah satu atau kedua conditional tak terpenuhi maka akan return false.



## SOAL 2

```
lat_2.py > cek_digit_belakang
1 def cek_digit_belakang(a,b,c):
2     bil_1=a%10
3     bil_2=b%10
4     bil_3=c%10
5     if bil_1==bil_2 or bil_1==bil_3 or bil_2==bil_3:
6         return True
7     else: return False
8
9 a=int(input("masukan bilangan pertama: "))
10 b=int(input("masukan bilangan kedua : "))
11 c=int(input("masukan bilangan ketiga : "))
12 print (cek_digit_belakang(a,b,c))
13
```

```
masukan bilangan pertama: 28
masukan bilangan kedua : 148
masukan bilangan ketiga : 10008
True
```

```
SUS VivoBook/Downloads/LapPrak_4/1:
masukan bilangan pertama: 2
masukan bilangan kedua : 3
masukan bilangan ketiga : 1
False
```

Pada soal kedua kita diminta untuk mengecek apakah ada minimal 2 angka dari input user yang memiliki digit paling kanan yang sama jika ya maka akan mengeluarkan hasil true dan jika tidak maka akan mengeluarkan output false

Pada code ini saya membuat function dengan nama cek\_digit\_belakang(a,b,c) kemudian saya membuat 3 variable yang akan menghitung digit belakangnya dengan memakai rumus modulo (%) 10 dari digit inputan user setelah itu ketiga variable tersebut di cek memakai conditional if bil\_1==bil\_2 or bil\_1==bil\_3 or bil\_2==bil\_3 conditional ini mengecek apakah ada 2/lebih digit belakang yang sama diantara tiga bilangan ini jika iya maka akan menghasilkan True jika tidak maka False

### SOAL 3

```
lat_3.py > ...
1  # Buatlah fungsi-fungsi konversi suhu menggunakan lambda function. Fungsi-fungsi
2  # yang harus anda implementasikan:
3  # Celcius to Fahrenheit.  $F = (9/5) * C + 32$ 
4  # Celcius to Reamur.  $R = 0.8 * C$ 
5  # Berikan contoh penggunaannya untuk test-case berikut ini:
6  # Input C = 100. Output F = 212.
7  # Input C = 80. Output R = 64.
8  # Input = 0. Output F = 32.
9  convert = {
10     "Fahrenheit": lambda C: (9/5) * C + 32,
11     "F": lambda C: (9/5) * C + 32,
12     "Reamur" : lambda C: 0.8 * C,
13     "R": lambda C: 0.8 * C
14 }
15
16 C=input("masukan suhu dalam celcius:")
17 skala=input("ubah suhu ke dalam satuan Fahrenheit/Reamur (F/R): ")
18 try:
19     C=int(C)
20     print (convert[skala](C))
21 except:
22     print ("anda salah memasukan input")
23
```

```
PS C:\Users\ASUS VivoBook\Downloads\LapPrak_4> & "C:/Users/ASUS VivoBook/Downloads/LapPrak_4/lat_3.py"
masukan suhu dalam celcius:100
ubah suhu ke dalam satuan Fahrenheit/Reamur (F/R): F
212.0
PS C:\Users\ASUS VivoBook\Downloads\LapPrak_4>
```

Pada soal ketiga ini kita diminta membuat function untuk merubah suhu dari celcius menjadi Fahrenheit / Reamur memakai lambda function.

Pada code ini pertama tama saya membuat dictionary Bernama convert dan membuat beberapa kata kunci diantaranya ialah "Fahrenheit", "F", "Reamur", dan "R" pada kata kunci "F" dan "Fahrenheit" saya menuliskan fungsi konversi dari celcius ke Fahrenheit memakai lambda yang dituliskan "F": lambda C: (9/5) \* C + 32 sama dengan "Fahrenheit" lalu pada kata kunci "Reamur" dan "R" dituliskan "R": lambda C: 0.8 \* C

Setelah itu maka input user diperlukan untuk memasukan suhu dalam celcius kemudian memasukan skala nya untuk input skala dibatasi untuk kata kunci yg disediakan dalam dictionarynya oleh karena itu saya menetapkan penanganan kesalahan berupa try and except agar tidak erorr.

ketika input dari user sesuai ketentuan maka akan memprint function convert dengan keyword yang dimasukan dalam input skala dan parameter C yang akan mengeluarkan hasil sesuai ketentuan

jika ada salah input atau skala tidak sesuai dengan keyword maka akan keluar output berupa “anda salah memasukan input”.

[https://github.com/vianchr/LapPrak\\_4](https://github.com/vianchr/LapPrak_4)