LAPORAN PRAKTIKUM

PEMROGRAMAN KOMPUTER (PYTHON)

Disusun Oleh:
Rafli Pratama
H1101241008

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS TANJUNGPURA PONTIANAK
2024

Praktikum 6

Fungsi

1.1. Konsep Dasar Fungsi

1.1.1 Konsep Fungsi

Fungsi adalah salah satu operator dalam bahasa pemrograman python dalam mempermudah penulisan program yang menggunakan kode yang berulang-ulang dengan nilai yang berbeda- beda. Pada materi ini dibahas materi terkait pengenalan fungsi tanpa return dan dengan menggunakan return, Variable-length Argument, Default Argument, Keyword Argument pada Function, Keyword-length Argument pada Function, Variable Scope pada Python.

Fungsi merupakan blok dari kode yang dirancang untuk melakukan tugas khusus atau instruksi yang dieksekusi ketika dipanggil dari bagian lain dalam suatu program. Tujuan pembuatan fungsi adalah:
a. Memudahkan dalam pembuatan program
b. Menghemat ukuran program
c. Membuat program menjadi terstruktur
d. Mengurangi duplikasi kode
e. Fungsi dapat dipanggil dari program atau fungsi lainnya.

Keuntungan fungsi pada pemrograman antara lain:
a. Menguraikan tugas pemrograman yang rumit menjadi langkah-langkah yang lebih kecil dan sederhana.
b. Mengurangi duplikasi kode (kode yang sama ditulis berkali-kali pada modul) pada program.
c. Dapat menggunakan kode yang ditulis dalam berbagai program yang berbeda.
d. Memecah program besar menjadi lebih kecil, msehingga mempermudah pengerjaan program.
e. Meningkatkan kemampuan pelacakan kesalahan, jika ada kesalahan, kita hanya mencari fungsi yang bersangkutan saja.

Kategori Fungsi :

- Standard Library Function merupakan fungsi-fungsi yang disediakan oleh Interpreter Python dalam file-file atau librarynya.
  Misalnya: input(), print(), open(), len(), max(), min(), abs() dll.
- Programme-Defined Function merupakan fungsi yang dibuat oleh programmer itu sendiri. Fungsi ini memiliki nama tertentu yang unik dalam sebuah program, posisinya terpisah dari program utama, dan bisa dijadikan satu ke dalam suatu library buatan programmer itu sendiri.

1.1.2 Deklarasi dan Pemakaian Fungsi

Pada umumnya fungsi memerlukan masukan yang disebut parameter atau argument. Hasil akhir fungsi akan berupa nilai balik fungsi. Bentuk umum sebuah fungsi adalah:
Fungsi pada Python dibuat dengan kata def lalu diikuti dengan nama fungsinya dan diikuti tanda semicolon. Seperti blok kode, kita memberikan identasi (tab) untuk menuliskan isi dari fungsi.
Contoh:

Untuk pemanggilan fungsi cukup menulis nama fungsinya saja
Maka output dari fungsi adalah.

```
def NamaFungsi():
    print ("Ini adalah Fungsi")
NamaFungsi()
```
✓ 0.0s

```
Ini adalah Fungsi
```

Fungsi dapat kita panggil lagi, dengam menulis nama fungsinya. Misal, dipanggil fungsi sebanyak 4 kali :
Maka outputnya adalah

```
def selamat():
    print ("Selamat belajar fungsi pada python")
selamat()
selamat()
selamat()
selamat()
```
✓ 0.0s

```
Selamat belajar fungsi pada python
Selamat belajar fungsi pada python
Selamat belajar fungsi pada python
Selamat belajar fungsi pada python
```

Penulisan kode program (sintaks) fungsi ada dua jenis yaitu

a. Menggunakan parameter
b. Tanpa parameter

Fungsi tanpa parameter sifatnya tetap dan tidak ada perubahan nilai pada saat pemanggilan fungsi tersebut.
Berikut adalah program menggunakan parameter seperti berikut ini :

```
def fungsi (parameter):
    print(parameter)
fungsi(1)
```
✓ 0.0s

```
1
```

Contoh program dengan penggunaan parameter adalah :

```
def selamat (kata):
    print (kata)
selamat("Penggunaan parameter pada python")
```
✓ 0.0s

```
Penggunaan parameter pada python
```

Kata "Penggunaan parameter pada python "adalah nilai parameter yang kita berikan.
Berikut adalah program tanpa parameter seperti berikut ini :

```
def fungsi():
    print()
fungsi()
```
✓ 0.0s

Contoh program tanpa penggunaan parameter adalah.

```
# Fungsi tanpa parameter dan nilai
def dataMhs():
    print ('NIM saya adalah H111202002')
    print ('Nama saya adalah Budi')
    print ('Berasal dari PS Sisfo FMIPA')
dataMhs()
```

✓ 0.0s

```
NIM saya adalah H111202002
Nama saya adalah Budi
Berasal dari PS Sisfo FMIPA
```

Operator Fungsi dapat juga memanggil fungsi lainnya, dengan kata lain fungsi memanggil fungsi lain. Contoh dapat dilihat pada program berikut ini:

```
def cek():
    q = int(input ('Masukkan angka :'))
    if q%2 == 0: print ('Angka', q, 'yang Anda Masukkan adalah GENAP')
    else: print ('Angka', q, 'yang Anda Masukkan adalah GANJIL')
def jilnap():
    for i in range (4):
        cek()
print("PROGRAM MENENTUKAN BILANGAN GANJIL GENAP")
jilnap()
```

✓ 8.1s

```
PROGRAM MENENTUKAN BILANGAN GANJIL GENAP
Angka 4 yang Anda Masukkan adalah GENAP
Angka 3 yang Anda Masukkan adalah GANJIL
Angka 2 yang Anda Masukkan adalah GENAP
Angka 1 yang Anda Masukkan adalah GANJIL
```

Berikut adalah contoh yang menggunakan parameter :

```
def cek(qw):
    if qw%2 == 0:
        print ('Angka', qw, 'yang Anda Masukkan adalah GENAP')
    else:
        print ('Angka', qw, 'yang Anda Masukkan adalah GANJIL')
def ulang():
    for i in range(4):
        d = int(input ('Masukkan angka :'))
        cek(d)
print("PROGRAM MENENTUKAN BILANGAN GANJIL GENAP")
ulang()
```

✓ 4.7s

```
PROGRAM MENENTUKAN BILANGAN GANJIL GENAP
Angka 1 yang Anda Masukkan adalah GANJIL
Angka 2 yang Anda Masukkan adalah GENAP
Angka 3 yang Anda Masukkan adalah GANJIL
Angka 4 yang Anda Masukkan adalah GENAP
```

Jika pada fungsi ulang yang memanggil fungsi cek kita gunakan parameter "qw", maka nilai ini tidak dikenali oleh fungsi cek. Karena parameter "qw" hanya dikenali pada fungsi cek yang pertama. Ini yang disebut dengan variabel lokal dan variabel global.

1.1.3 Fungsi dengan lebih dari 1 parameter

a) Parameter wajib
   Berikut adalah contoh lebih dari 1 parameter yang wajib diisi :

```
def luassegitiga (a,t):
    luas=a*t/2
    print("maka luas segitiginya adalah", luas, "m2")
def keliling(a):
    j=3*a
    print("maka keliling segitiga adalah ", j, "m2")

print("PROGRAM MENCARI LUAS DAN KELILING SEGITIGA SAMA SISI ")
a1=float(input("input sisi alasnya: "))
t1=float(input("input tingginya : "))
luassegitiga (a1,t1)
keliling(a1)
```

✓ 4.1s

```
PROGRAM MENCARI LUAS DAN KELILING SEGITIGA SAMA SISI
maka luas segitiginya adalah 12.0 m2
maka keliling segitiga adalah  12.0 m2
```

Fungsi luassegitiga memiliki 2 parameter yaitu a dan t. 2 parameter ini hanya dikenali pada blok fungsi saja. Ketika di panggil pada program utama memanggil fungsi luassegitiga menggunakan parameter yang berbeda yaitu al, dan t1 yaitu variabel global. Variabel ini bisa digunakan untuk dipanggil ke dalam fungsi lainnya yaitu untuk memanggil fungsi keliling.

b) Parameter opsional

Tidak semua parameter fungsi pada python itu bersifat wajib. Ada yang opsional. Contoh berikut ini :

```
def kota (daerah, derajat, satuan = 'celcius'):
    print(f"Suhu di daerah {daerah} adalah {derajat} {satuan}")
kota ("pontianak", 28)
kota ("pontianak", 18, "kelvin")
```

✓ 0.1s

```
Suhu di daerah pontianak adalah 28 celcius
Suhu di daerah pontianak adalah 18 kelvin
```

Pada fungsi kota () tersebut, kita mendefinisikan 3 buah parameter yaitu daerah, derajat, dan suhu = 'celcius'. Dua parameter pertama bersifat wajib di isi dan parameter ketiga tidak wajib diisi, tapi jk tdk isi maka akan mengeluarkan defaulnya "celcius", tapi jika diisi dengan parameter lain seperti "kelvin" maka parameter satuannya akan dicetak "kelvin".

1.1.4 Ruang Lingkup Variabel

Ruang lingkup variabel adalah keadaan pendeklarasian sebuah variabel di tentukan. Secara umum terdapat 2 ruang lingkup variabel pada python yaitu :

a) Variabel lokal dan,
b) Variabel global

Variabel lokal adalah variabel yang didefinisikan didalam sebuah fungsi (def). Artinya variabel tersebut hanya dapat di gunakan dalam cakupan fungsi tersebut saja. Dan jika sebuah variabel didefinisikan diluar fungsi maka variabel tersebut bersifat global. Variable global adalah variabel yang bisa dipanggil dari manapun dari satu file python. Perhatikan contoh berikut:

```
def fungsiSatu():
    x = 'Rafli'
    print(f'Nilai dalam fungsi : {x}')
def fungsiDua():
    global x
    x = 'Sidiq'
    print(f'Nilai dalam fungsi adalah : {x}')

x = 'Poundra'

print(f'Nilai sebelum fungsi : {x}')
fungsiSatu()
print(f'Nilai sesudah fungsi : {x}')
print()

print(f'Nilai sebelum fungsi : {x}')
fungsiDua()
print(f'Nilai sesudah fungsi : {x}')
```

```
Nilai sebelum fungsi : Poundra
Nilai dalam fungsi : Rafli
Nilai sesudah fungsi : Poundra

Nilai sebelum fungsi : Poundra
Nilai dalam fungsi adalah : Sidiq

Nilai sesudah fungsi : Sidiq
```

1.1.5 Fungsi Dengan Return

Fungsi dapat ditambahkan pernyataan/statement ". Pernyataan ini digunakan untuk mengembalikan suatu nilai ke badan program yang memanggil fungsi. Fungsi dari segi pengembalian nilai dibagi menjadi 2 yaitu, fungsi yang tidak mengembalikan nilai dan fungsi yang mengembalikan nilai. Contoh-contoh program sebelumnya adalah fungsi yang tidak mengembalikan nilai.
Berikut adalah contohnya :

```
def luassegitiga ():
    a = int(input ('Masukkan alas: '))
    t = int(input ('Masukkan tinggi: '))
    L =a*t/2
    return (L)
def utama():
    luas = luassegitiga()
    print ('Luas segitiga adalah = ',luas)
print ("Menghitung Luas Segitiga")
utama()
```

✓ 2.9s

```
Menghitung Luas Segitiga
Luas segitiga adalah =  12.0
```

Sebagai contoh program untuk menghitung luas persegi panjang, dengan dua function luaspersegipanjang dan utama. Dalam blok function luaspersegipanjang terdapat statemen return yang akan mengembalikan nilai L (luas persegi panjang) ke badan function utama yang telah memanggil function luaspersegipanjang. Berikut struktur program untuk menghitung luas persegi panjang.

1.1.6 Statetement Lambda

Python juga menyediakan bentuk fungsi yang lain yaitu "lambda". Lambda adalah fungsi yang tidak memiliki nama atau disebut fungsi anonim (anonymous function). Fungsi Lambda dapat memiliki banyak argumen dengan satu ekspresi. Lambda bukanlah sebuah perintah atau statemen namun merupakan sebuah ekspresi.

Fungsi argumen sama dengan parameter pada def, argumen bisa lebih dari satu, dan diikuti oleh ekspresi setelah tanda titik dua (:). Setiap fungsi anonim yang didefinisikan dengan Python memiliki 3 bagian yaitu:

Kata kunci lambda,
Parameter (atau variabel terikat),
Badan fungsi.

Lambda dapat memiliki beberapa parameter, tetapi badan fungsi hanya berisi satu ekspresi. Saja. Selain itu, lambda ditulis dalam satu baris kode dan juga dapat segera dipanggil. Berikut ini adalah contoh syntax lambda Python:

Kode program:
lambda argument1, argument2,..., argumentN, espression using arguments
Fungsi argumen sama dengan parameter pada def, argumen bisa lebih dari satu, dan diikuti oleh ekspresi setelah tanda titik dua (:).

Perbedaan lambda dan def dapat dilihat pada contoh berikut ini:
Dalam bentuk def:

```
def Fungsi():
    return "Hello World"
fungsi()
```

Dalam bentuk lambda, maka penulisanya lebih sederhana.

```
  lambda: "Hello World"
✓ 0.0s
<function __main__.<lambda>()>
```

Untuk memanggil fungsi lambda, tidak dapat memanggil secara langsung. Kita harus menggunakan variabel lain atau memanggil dalam bentuk fungsi. Berikut contoh penggunaan bentuk lambda,

```
x = lambda a,b,c: (a+b)-(b-c)
x(8,6,2)
✓ 0.0s
10
```

Atau untuk melihat perbedaan def dengan lambda dapat dilihat ilustrasi berikut ini.

```
#fungsi def
def f(x):
    return x*4
print (f(10))
#fungsi lambda
g = lambda x: x*4
print (g(15))
✓ 0.0s
40
60
```

Lambda tidak menyertakan pernyataan "return" atau kembali, lambda pasti akan mengembalikan nilai fungsi. Tapi, lambda harus menggunakan variabel lain untuk memanggil fungsinya.
Contoh lain yang sebagai pengayaan adalah :

```
w= lambda q,r: q*r/2
q=int(input("alas"))

r=int(input("tinggi"))
w(q,r)

print("nilai w adalah", w)
```
✓ 1.3s

```
nilai w adalah <function <lambda> at 0x0000021BB6CDF2E0>
```

Jelaskan kenapa tidak bisa menampilkan hasil perhitngan luas segitiga. Padahal lambda secara default akan mengembalikan ke fungsinya (tidak perlu return).

1.1.7 Fungsi Rekursif

Rekursif adalah salah satu metode dalam dunia matematika, merupakan sebuah fungsi yang mengandung fungsi itu sendiri. Dalam dunia pemrograman, rekursif diimplementasikan dalam sebuah fungsi yang memanggil dirinya sendiri, sehingga terjadilah perulangan didalam fungsi tersebut. Pada proses rekursif, akan terjadi secara berulang-ulang. Oleh karena itu, perlu adanya stopping role atau penyetopan untuk penghentian proses perulangan tersebut.
Kelebihan

a) Program lebih singkat.
b) Pada beberapa kasus, lebih mudah menggunakan fungsi rekursif, contohnya: pangkat, factorial, dan fibonacci, dan beberapa proses deret lainnya.
c) Lebih efisien dan cepat dibandingkan proses secara iteratif.

Kekurangan

a) Memakan memori lebih besar, karena setiap bagian dari dirinya dipanggil, akan membutuhkan sejumlah ruang memori untuk penyimpanan.
b) Rekursif sering kali tidak bisa berhenti sehingga memori akan terpakai habis dan program bisa hang.

Sebagai ilustrasi fungsi rekursif yaitu deret fibonacci. Barisan yang berawal dari 0 dan 1, kemudian angka berikutnya didapat dengan cara menambahkan kedua bilangan yang berurutan sebelumnya. Dengan aturan ini, maka barisan bilangan Fibonaccci yang pertama adalah: 0, 1,
1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, ... dan seterusnya.

```
# fungsi rekursif pada deret fibonacci
def fibo (n):
    if n<=2:
        return 1
    else:
        return (fibo(n-1)+fibo(n-2))
n = int(input ('Masukkan nilai n: '))
print ("Berikut adalah tampilan Deret Fibonacci sebanyak", n)
for i in range (n):
    print (fibo (i), end=" ")
```
✓ 1.3s

```
Berikut adalah tampilan Deret Fibonacci sebanyak 5
1 1 1 2 3
```

Selain deret fibonacci ada juga contoh fungsi rekursif yaitu pangkat.

```
def pangkat(a,b):
    if b == 0:
        return 1
    else:
        return a * pangkat (a,b-1)
a = int(input("Masukan Angka: "))
b = int(input("Masukan pangkanya: "))
print(a," pangkat", b, "adalah" ,pangkat (a,b))
```

✓ 2.4s

```
2  pangkat 2 adalah 4
```

Fungsi pangkat() bertujuan menghitung bilangan a yang dipangkatkan dengan bilangan b. Fungsi tersebut terdapat pemanggilan dirinya sendiri, yang disebut rekursif. Setiap nilai a dan b yang di masukan user dikirim ke fungsi pangkat() melalui parameter variabel a dan b.
Selama nilai b bukan 0 maka fungsi pangkat() akan terus memangil dirinya sendiri, dan nilai b akan terus di kurangi 1 sampai kondisi bisa terpenuhi dan perulangan dihentikan.

1.1.8 Keyword Argumen Pada Function

```
def luasSegitiga(a, t):
    L = (a * t) / 2
    print('Luas = (%d x %d) / 2' % (a, t))
    print('= %.2f' % L)

def hitung():
    alas = float(input('Alas = '))
    tinggi = float(input('Tinggi = '))
    luasSegitiga(alas, tinggi)

hitung()
```

✓ 2.1s

```
Luas = (4 x 4) / 2
= 8.00
```

Pada program diatas, saat akan memanggil function luasSegitiga dengan parameter, nilai parameter yang dituliskan harus sesuai urutan parameter pada function. Seperti untuk memanggil function luasSegitiga luasSegitiga (alas, tinggi), alas yang dituliskan di urutan pertama akan dikonversi ke parameter a yang berada di urutan pertama pada functionluas Segitiga, dan tinggi yang dituliskan di urutan kedua dikonversi ke parameter t yang berada pada urutan kedua pula. Hal ini mungkin cukup mudah apabila jumlah parameter sedikit, Namun, bagaimana jika parameter yang dibutuhkan banyak dan berulang, hal tersebut mungkin akan sangat menyulitkan, dikarenakan kita harus mengingat posisi masing-masing parameter. Oleh karena itu, solusi untuk mengatasi masalah tersebut, di dalam python pada saat memanggil function gunakan statemen kata kunci (keyword) untuk menuliskan parameter di badan function yang dipanggil, sehingga penulisan parameter tidak perlu menyesuaikan urutan. Sebagai contoh program dengan statemen yang menggunakan kata kunci pada saat memanggil function.

Sebagai contoh program lain untuk melihat perbedaan statemen yang menggunakan keyword dengan statemen yang tanpa keyword.

```
def data_siswa (nama, kelas, jurusan):
    print ("nama: ", nama)
    print( "kelas:", kelas)
    print ("jurusan: ", jurusan)
    print ("Menggunakan keyword parameter:")
data_siswa (kelas="XII", jurusan="IPA", nama="Zahwa")
print (f"\nTanpa menggunakan keyword parameter,\n urutan acak:")
data_siswa ("XII", "IPA", "Zahwa")
print(f"\nTanpa menggunakan keyword parameter, urutan sesuai :" )
data_siswa ("Zahwa", "XII", "IPA")
```

✓ 0.0s

```
nama:  Zahwa
kelas: XII
jurusan:  IPA
Menggunakan keyword parameter:

Tanpa menggunakan keyword parameter,
 urutan acak:
nama:  XII
kelas: IPA
jurusan:  Zahwa
Menggunakan keyword parameter:

Tanpa menggunakan keyword parameter, urutan sesuai :
nama:  Zahwa
kelas: XII
jurusan:  IPA
Menggunakan keyword parameter:
```

1.1.9 Keyword-length Argument pada Function

Di dalam function, keyword pada parameter (argumen) dapat memiliki jumlah yang banyak dan nilai yang berubah-ubah. Agar nilai-nilai pada keyword berbentuk tuple, maka penulisan parameter harus diawali dengan lambang satu bintang (*). Sedangkan untuk nilai-nilai yang berbentuk dictionary, maka penulisan parameter harus diawali dengan lambang dua bintang (**). Sebagai contoh perhatikan berikut ini:

```
#lambang dikenali sebagai tuple
#lambang ** dikenali sebagai dictionary
def biodata (nama, usia, *hobi, **lain2):
    print ("Nama: ",nama)
    print ("Usia: ",usia)
    print ("Hobi: ",hobi)
    print ("Lain-lain: ", lain2 )
biodata ("Rafa", "7", "Membaca", "Futsal", "Berenang", Jenis_kelamin="Laki-Laki", status="Belum Menikah")
```

✓ 0.0s

```
Nama:  Rafa
Usia:  7
Hobi:  ('Membaca', 'Futsal', 'Berenang')
Lain-lain:  {'Jenis_kelamin': 'Laki-Laki', 'status': 'Belum Menikah'}
```

1.2. Tugas Praktikum

1.2.1. Soal 1

Buatlah program dalam bentuk fungsi menu, fungsi untuk mencari luas bangun datar dan fungsi untuk mencari volume bangun ruang!

```python
def menu():
    operasi = int(input('Silahkan pilih Operasi : \n1. Mengitung Luas Bangun datar \n2. Menghitung Volume Bangun ruang\n'))
    return operasi

def luasBangunDatar():

    def persegi():
        a = int(input('Masukkan sisi a: '))
        b = int(input('Masukkan sisi b: '))
        luas = a * b
        print(f'Luas persegi dengan sisi {a} cm dan {b} cm adalah {luas} cm2')
        return luas

    def persegiPanjang():
        a = int(input('Masukkan panjang: '))
        b = int(input('Masukkan lebar: '))
        luas = a * b
        print(f'Luas persegi panjang dengan panjang {a} cm dan lebar {b} cm adalah {luas} cm2')
        return luas

    def segitiga():
        a = int(input('Masukkan sisi alas: '))
        t = int(input('Masukkan tinggi: '))
        luas = (a * t) / 2
        print(f'Luas segitiga dengan alas {a} cm dan tinggi {t} cm adalah {luas} cm2')
        return luas

    bangunDatar = int(input('Silahkan pilih bangun datar:\n1. Persegi\n2. Persegi panjang\n3. Segitiga'))

    if bangunDatar == 1:
                persegi()
    elif bangunDatar == 2:
                persegiPanjang()
    elif bangunDatar == 3:
                segitiga()
    else:
                print('Silahkan masukkan pilihan yang benar')
```

```python
def volumeBangunRuang():
    bangunRuang = int(input('Silahkan pilih bangun datar:\n1. Kubus\n2. Balok'))
    def kubus():
        a = int(input('Masukkan sisi a: '))
        volume = a * a * a
        print(f'Volume kubus dengan sisi {a}cm adalah {volume} cm3')
        return volume
    def balok():
        p = int(input('Masukkan panjang: '))
        l = int(input('Masukkan lebar: '))
        t = int(input('Masukkan tinggi : '))
        volume = p*l*t
        print(f'Volume balok dengan panjang{p} cm lebar {l}cm dan tinggi {t}cm adalah {volume}cm3')
        return volume

    if bangunRuang == 1:
        kubus()
    elif bangunRuang == 2:
        balok()
    else:
        print('Silahkan masukkan pilihan yang benar')

operasi = menu()
if operasi == 1:
    print('Anda memilih opsi 1. menghitung luas bangun datar, Silahkan pilih bangun datar yang anda inginkan')
    luasBangunDatar()
elif operasi == 2:
    print('Anda memilih opsi 1. menghitung luas bangun ruang, Silahkan pilih bangun ruang yang anda inginkan')
    volumeBangunRuang()
else:
    print('Silahkan masukkan pilihan operasi yang tepat')
```

✓ 5.2s

```
Anda memilih opsi 1. menghitung luas bangun ruang, Silahkan pilih bangun ruang yang anda inginkan
Volume kubus dengan sisi 4cm adalah 64 cm3
```

1.2.1. Soal 2

Buatlah dengan menggunakan fungsi sebuah algoritma untuk menebak bilangan acak dengan ketetentuan sebagai berikut:

- Terdapat fungsi membuat bilangan acak dari 2 parameter nilai batas bawah dan batas atas berdasarkan masukan dari user
- Terdapat fungsi untuk menebak bilangan dengan seleksi tebakan kebesaran, tebakan kekecilan dan tebakan benar
- Salah satu output dapat menghitung jumlah tebakan sampai tebakan benar

```python
import random

def acak():
    a = int(input('Masukkan batas angka awal : '))
    b = int(input('Masukkan batas angka akhir : '))
    print('Tebak Angka!!!')
    print(f'Silahkan Tebak angka yang benar antara {a} dan {b}')
    bilanganAcak = random.randint(a,b)
    return bilanganAcak

def tebakAngka ():
    tebakan = 1
    while True:
        print(f'----------Tebakan ke -{tebakan}----------')
        angka = int(input('Masukkan Angka'))
        print(f'Tebakan Anda adalah angka {angka}')
        if angka == bilanganAcak:
            print(f'Tebakan anda benar yaitu angka {bilanganAcak}')
            tebakan = 1
            break
        elif angka < bilanganAcak :
            print(f'Tebakan anda salah, selisihnya adalah {bilanganAcak - angka}')
            tebakan +=1
        elif angka > bilanganAcak:
            print(f'Tebakan anda salah, selisihnya adalah {angka - bilanganAcak}')
            tebakan +=1

bilanganAcak = acak()
tebakAngka()
```

Output :

```
Tebak Angka!!!
Silahkan Tebak angka yang benar antara 5 dan 25
----------Tebakan ke -1----------
Tebakan Anda adalah angka 4
Tebakan anda salah, selisihnya adalah 15
----------Tebakan ke -2----------
Tebakan Anda adalah angka 19
Tebakan anda benar yaitu angka 19
```

Penjelasan :
Pertama tama menggunakan library random, lalu kita membuat fungsi yang menentukan nilai random setelah itu kita membaut fungsi yang membuat user dapat menebak bilangan dengan menginputkkan sebuah angka, lalu jika angka sama dengan angka random maka program selesai dan jika salah maka user akan di minta meninputkan kembali sampai tebakan nya benar

1.2.1. Soal 3

a. PT. Kalila hanya memberikan tunjangan tambahan untuk anak seorang karyawan dengan syarat maksimal sampai 3 anak. Setiap anak diberikan tunjangan sebesar 15% dari gaji pokok.
Tulis program untuk menghitung total tunjangan anak, bila gaji pokok dan jumlah anak diinput oleh user.

```python
Nama_Karyawan = []
Anak_Karyawan = []
Gaji_Karyawan = []
def dataKaryawan ():
    namaKaryawan = input('Masukkan nama karyawan : ')
    jumlahAnak = int(input('Masukkan Nama karyawan'))
    return namaKaryawan, jumlahAnak
def olahGaji():
    if jumlahAnak == 0:
        gaji = 3000000 + 0
    elif jumlahAnak == 1:
        gaji = 3000000 + 450000
    elif  jumlahAnak <= 3:
        gaji= 3000000 + 450000
    elif  jumlahAnak > 3 :
        gaji = 3000000 + 450000
    else :
        print('Data yang anda inputkan salah')
    return gaji

def listKaryawan ():
    Nama_Karyawan.append(namaKaryawan)
    Anak_Karyawan.append(jumlahAnak)
    Gaji_Karyawan.append(gaji)

def cetak():
    print(f'|\t  {Nama_Karyawan[i]}  \t|\t  {Anak_Karyawan[i]}  \t\t|\t  {Gaji_Karyawan[i]}  \t|')
    print('-------------------------------------------------------------------------')

data = int(input('Masukkab jumlah karyawan : '))
print('-------------------------------------------------------------------------')
print(f'|\tNama Karyawan\t|\tJumlah Anak\t|\tGaji Karyawan\t|')
print('-------------------------------------------------------------------------')
for i in range (data):
    namaKaryawan, jumlahAnak = dataKaryawan()
    gaji = olahGaji()
    listKaryawan()
    cetak()
```

Output :

```
-------------------------------------------------------------------------
|       Nama Karyawan   |       Jumlah Anak     |       Gaji Karyawan   |
-------------------------------------------------------------------------
|         Rafli         |         0             |         3000000       |
-------------------------------------------------------------------------
|         Atha          |         3             |         3450000       |
-------------------------------------------------------------------------
|         Audy          |         5             |         3450000       |
-------------------------------------------------------------------------
```

b. Sebuah perusahaan PT. Indo Makmur menggaji karyawannya secara mingguan dengan hitungan sebagai berikut:
golongan 1 dengan upah per jam 3.000 rupiah
golongan 2 dengan upah per jam 3.500 rupiah
golongan 3 dengan upah per jam 4.000 rupiah
golongan 4 dengan upah per jam 5.000 rupiah
Bila seorang karyawan bekerja kurang atau sama dengan 40 jam per minggu, akan dihitung dengan upah per jam seperti di atas, tetapi apabila bekerja lebih dari 40 jam, maka lebihnya akan dihitung sebagai lembur dengan upah per jam 1½ kali upah biasa. Tulis algoritma dan flowchartnya untuk menghitung gaji mingguan karyawan, bila golongan dan jam kerja diinput dari keyboard

```python
Nama_Karyawan = []
Golongan_Karyawan = []
Jam_Kerja_Karyawan = []
Gaji_Karyawan = []

def dataKaryawan():
    namaKaryawan = input('Masukkan nama karyawan: ')
    jamKerjaKaryawan = int(input('Masukkan Jam Kerja karyawan: '))
    golongan = int(input('Masukkan Golongan karyawan (1-4): '))
    return namaKaryawan, jamKerjaKaryawan, golongan

def olahGaji(jamKerjaKaryawan, golongan):
    if jamKerjaKaryawan > 40:
        jamLembur = jamKerjaKaryawan - 40
        jamKerjaKaryawan = 40
    else:
        jamLembur = 0

    if golongan == 1:
        gaji = jamKerjaKaryawan * 3000 + (3000 * 1.5 * jamLembur)
    elif golongan == 2:
        gaji = jamKerjaKaryawan * 3500 + (3500 * 1.5 * jamLembur)
    elif golongan == 3:
        gaji = jamKerjaKaryawan * 4000 + (4000 * 1.5 * jamLembur)
    elif golongan == 4:
        gaji = jamKerjaKaryawan * 5000 + (5000 * 1.5 * jamLembur)
    else:
        gaji = 0  # Default in case of invalid golongan

    return gaji

def listKaryawan(namaKaryawan, jamKerjaKaryawan, golongan, gaji):
    Nama_Karyawan.append(namaKaryawan)
    Jam_Kerja_Karyawan.append(jamKerjaKaryawan)
    Golongan_Karyawan.append(golongan)
    Gaji_Karyawan.append(gaji)

def cetak(i):
    print(f'|\t{Nama_Karyawan[i]}\t\t|\t\t{Golongan_Karyawan[i]}\t\t|\t{Jam_Kerja_Karyawan[i]}\t\t|\t{Gaji_Karyawan[i]}\t|')
    print('-----------------------------------------------------------------------------------------------------------')

data = int(input('Masukkan jumlah karyawan: '))
print('-----------------------------------------------------------------------------------------------------------')
print(f'|\tNama Karyawan\t|\tGolongan Karyawan\t|\tJam Kerja\t|\tGaji Karyawan\t|')
print('-----------------------------------------------------------------------------------------------------------')

for i in range(data):
    namaKaryawan, jamKerjaKaryawan, golongan = dataKaryawan()
    gaji = olahGaji(jamKerjaKaryawan, golongan)
    listKaryawan(namaKaryawan, jamKerjaKaryawan, golongan, gaji)
    cetak(i)
```

Output :

```
-----------------------------------------------------------------------------------------------------------
|	Nama Karyawan	|	Golongan Karyawan	|	Jam Kerja	|	Gaji Karyawan	|
-----------------------------------------------------------------------------------------------------------
|	Rafli		|		4		|	40		|	200000.0	|
-----------------------------------------------------------------------------------------------------------
|	Atha		|		3		|	35		|	140000.0	|
-----------------------------------------------------------------------------------------------------------
|	Audy		|		2		|	45		|	166250.0	|
-----------------------------------------------------------------------------------------------------------
|	Arifqu		|		1		|	60		|	210000.0	|
-----------------------------------------------------------------------------------------------------------
```

1.3.1. Kesimpulan

Dari praktikum ini dapat disimpulkan :

1. Function dalam python memiliki dua jenis kode def dan lambda
2. Di dalam function dikenal kode return yang berfungsi untuk mengembalikan nilai ke dalam badan program yang memanggil function

3. Parameter pada function dapat didefinisikan ataupun tidak didefinisikan. Penulisan parameter untuk memanggil function jika sesuai urutan tidak menggunakan keyword, sebaliknya apabila dituliskan tidak sesuai dengan urutan maka harus menggunakan keyword
4. Salah satu penerapan function adalah untuk menyelesaikan fungsi rekursif
5. Scope variabel atau cakupan variabel merupakan suatu keadaan dimana pendeklarasian sebuah variabel di tentukan. Dalam scope variabel dikenal dua istilah yaitu local dan global.
6. Lambang yang digunakan untuk mendefinisikan argumen pada parameter berbentuk tuple dan dictionary adalah satu bintang (*) dan dua bintang (**)

1.3.2. Saran

Saran terhadap praktikum ini yaitu:

1. Hati hati saat menggunakan variabel scope, pastikan bisa membedakan variabel lokal dan global

DAFTAR PUSTAKA

Modul *Perkuliahan dan Praktikum Algoritma dan Pemrograman* Oleh Ilhamsyah, S.Si., M.Cs.