

✧ Kumpulan Hasil Praktikum Pemrograman Berbasis Fungsi



hello! I'm...

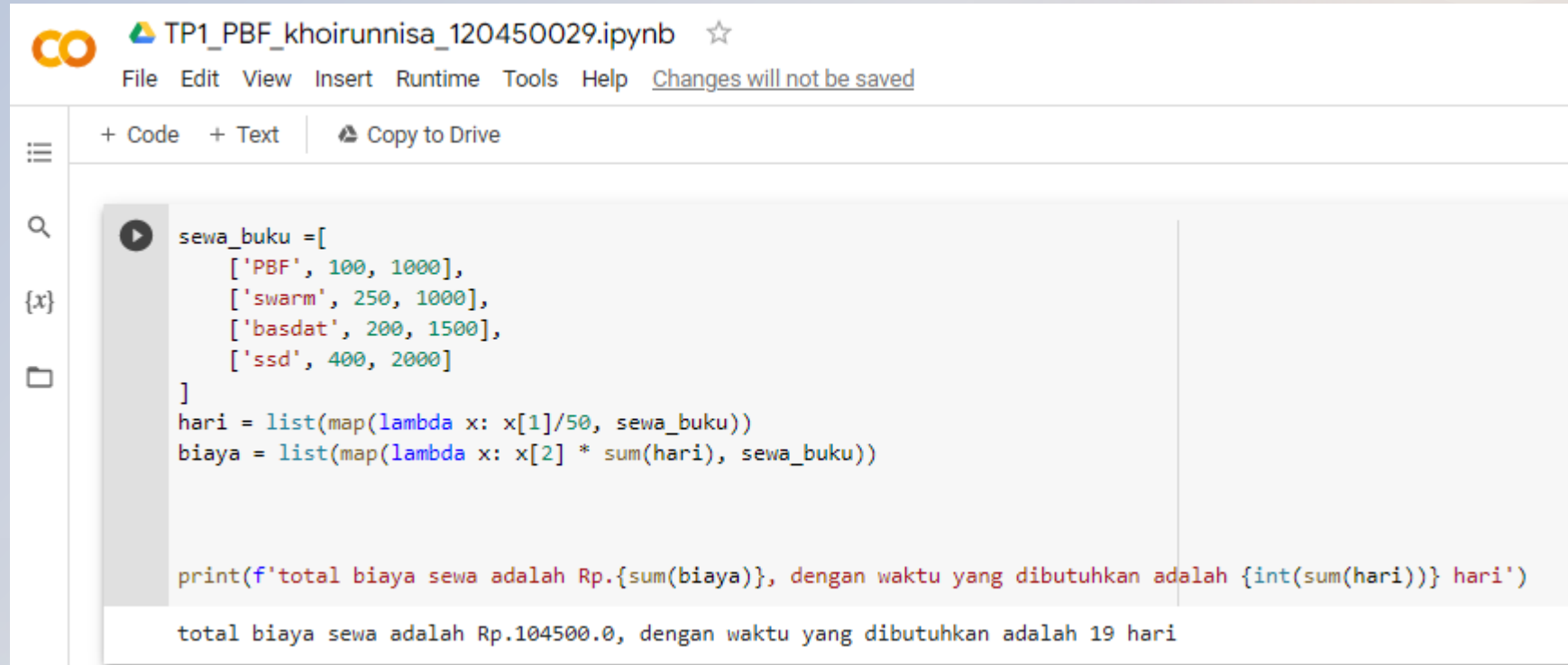
KHOIRUNNISA - 120450029



table of contents

- Tugas Pendahuluan 1
- Jurnal 1
- Tugas Pendahuluan 2
- Jurnal 2
- Tugas minggu 9
- Tugas minggu 10
- Tugas Minggu 11
- Tugas Minggu 12
- Tugas Minggu 13

Tugas Pendahuluan 1



The screenshot shows a Jupyter Notebook titled "TP1_PBF_khoirunnisa_120450029.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with options like "+ Code", "+ Text", and "Copy to Drive". The code cell contains the following Python code:

```
sewa_buku = [
    ['PBF', 100, 1000],
    ['swarm', 250, 1000],
    ['basdat', 200, 1500],
    ['ssd', 400, 2000]
]
hari = list(map(lambda x: x[1]/50, sewa_buku))
biaya = list(map(lambda x: x[2] * sum(hari), sewa_buku))

print(f'total biaya sewa adalah Rp.{sum(biaya)}, dengan waktu yang dibutuhkan adalah {int(sum(hari))} hari')
```

The output of the code is displayed below the cell:

```
total biaya sewa adalah Rp.104500.0, dengan waktu yang dibutuhkan adalah 19 hari
```

Nama : KHOIRUNNISA

NIM : 120450029

Kelas : RA

1 Pengantar Pemrograman Fungsional

1.1 Soal

Seorang mahasiswa sains data ingin menyewa buku dari sebuah startup yang menyediakan layanan sewa buku. Startup tersebut memiliki ketentuan sewa dengan aturan sebagai berikut: a. Harga sewa buku berbeda-beda sesuai dengan kategorinya b. Harga sewa buku dihitung berdasarkan jumlah halaman nya c. Harga sewa buku dihitung per hari nya d. Maksimal durasi sewa adalah 26 hari

Startup tersebut masih dalam tahap awal pengembangan, sehingga ingin melakukan uji coba penyewaan 5 kategori buku. Berikut rincian kategori nya:

- Kategori 1 : 100 rupiah per lembar per hari
 - Kategori 2 : 200 rupiah per lembar per hari
 - Kategori 3 : 250 rupiah per lembar per hari
 - Kategori 4 : 300 rupiah per lembar per hari
 - Kategori 5 : 500 rupiah per lembar per hari
- Startup tersebut memerlukan sebuah program untuk:
- menghitung total biaya dari customer
 - mencatat tanggal awal sewa, dan durasi hari
 - menampilkan informasi kapan tanggal pengembalian buku dari customer
- Format input tanggal adalah yyyy-mm-dd Bantulah startup tersebut membuat program tersebut dengan menggunakan konsep modularisasi!

Jurnal 1

```
Tanggal = input('Tanggal Pinjam: ')
durasi = int( input('Durasi Pinjam: ') )
```

```
kategori = {
    1 : 100,
    2 : 200,
    3 : 300,
    4 : 400,
    5 : 500
}
```

```
Tanggal Pinjam: 2022-01-02
Durasi Pinjam: 25
```

```
def dtl(s_tgl):
```

```
    return [ int(k) for k in s_tgl.split('-') ]
```

```
def is_cm(tgl_p,d,c):
    return tgl_p[2] + d > c
```

```
def thn_back(tgl_p,d,c):
    return tgl_p[0]+1 if ( is_cm(tgl_p,d,c) and tgl_p[1] == 12) else tgl_p[0]
```

```
def bln_back(tgl_p,d,c):
    return(tgl_p[1] % 12)+1 if is_cm(tgl_p,d,c) else tgl_p[1]
```

```
def tgl_back(tgl_p,d,c):
    return tgl_p[2] +d - c if is_cm(tgl_p,d,c) else tgl_p[2] + d
```

```
def is_awal_abad(thn):
    return thn % 100 == 0
```

```
    return thn % 100 == 0
```

```
def kabisat(thn):
    return ( is_awal_abad(thn) and thn % 400 == 0 ) or ( not is_awal_abad(thn) and thn % 4 ==
```

```
def dec_c(t):
    return 30 + ( t[1] % 2 if t[1] <= 8 else abs( (t[1] % 2)-1 ) ) if t[1]!= 2 else (29 if ka
```

```
def wkt_kembali(tgl_p,d):
    return [ thn_back( tgl_p,d, dec_c(tgl_p) ) , bln_back(tgl_p,d,dec_c(tgl_p)) , tgl_back(t
```

```
# Tanggal Pinjam: '2022-2-27'
```

```
# Durasi Pinjam: 1
```

```
tgl_p = dtl(Tanggal)
```

```
wkt_kembali(tgl_p,durasi)
```

```
[2022, 1, 27]
```

```
sewaan_all = [ [1,5], [2,3], [3,0], [4,1], [5,2]]
```

```
def calc_biaya_per_kategori(kategori,sewaan):
    return sewaan[1] * kategori.get( sewaan[0] )
```

```
def calc_all_biaya(kategori,sewaan_all,durasi):
    return sum( [calc_biaya_per_kategori(kategori,sewaan) for sewaan in sewaan_all] ) * duras
```

```
calc_all_biaya(kategori,sewaan_all,durasi)
```

```
62500
```

Tugas Pendahuluan 2

 TP_PBF_M2_029_Khoirunnisa.ipynb ☆

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text  Copy to Drive

 Tugas Pendahuluan PBF Modul 2

 Khoirunnisa

 120450029

1. Buatlah sebuah fungsi bernama ulang_i_NIM, ulang_i memiliki input sebuah bilangan skalar a, dan mengeluarkan vektor 1xn dengan seluruh elemen nya adalah a !

```
[ ] from functools import reduce as r

a = [1,2,0,4,5,0,0,2,9]

def ulang_i_029(x):
    return x * 1
print(list(map( ulang_i_029, a)))

[1, 2, 0, 4, 5, 0, 0, 2, 9]
```

2. Buatlah deret bilangan sebagai berikut dengan input n sebagai panjang deret:

$1/2, -1/4, 1/8, \dots, (-1)^n n 1/2^n$

```
[ ] n = int(input('Masukkan Nilai: '))
a = list(range(1,n))
def deret(n):
    return (-1**n) * (1/(2**n))
print(list(map(deret,a)))
```

Masukkan Nilai: 12

$[-0.5, -0.25, -0.125, -0.0625, -0.03125, -0.015625, -0.0078125, -0.00390625, -0.001953125, -0.0009765625, -0.00048828125]$

3. Jumlahkan deret bilangan tersebut!

```
[ ] n = int(input('Masukkan Nilai: '))
a = list(range(1,n))
def deret(n):
    return (-1**n) * (1/(2**n))
print(list(map(deret,a)))

L = list(map(deret,a))

def add(a,b):
    res = a+b
    print('a:',a, 'b:',b, '-> a+b:',res)
    return res
print(r( add, L ))
```

Tugas Pendahuluan 2

Masukkan Nilai: 12

`[-0.5, -0.25, -0.125, -0.0625, -0.03125, -0.015625, -0.0078125, -0.00390625, -0.001953125, -0.0009765625, -0.00048828125]`

a: -0.5 , b: -0.25 -> a+b: -0.75

a: -0.75 , b: -0.125 -> a+b: -0.875

a: -0.875 , b: -0.0625 -> a+b: -0.9375

a: -0.9375 , b: -0.03125 -> a+b: -0.96875

a: -0.96875 , b: -0.015625 -> a+b: -0.984375

a: -0.984375 , b: -0.0078125 -> a+b: -0.9921875

a: -0.9921875 , b: -0.00390625 -> a+b: -0.99609375

a: -0.99609375 , b: -0.001953125 -> a+b: -0.998046875

a: -0.998046875 , b: -0.0009765625 -> a+b: -0.9990234375

a: -0.9990234375 , b: -0.00048828125 -> a+b: -0.99951171875

-0.99951171875

4. Sebuah DNA dimodelkan dalam sebuah string menjadi sequence TCGA dan disimpan ke dalam data :

<https://drive.google.com/file/d/18C1ylsTXrY9pglqqlhijoS8LYmcxdIjM/view?usp=sharing>

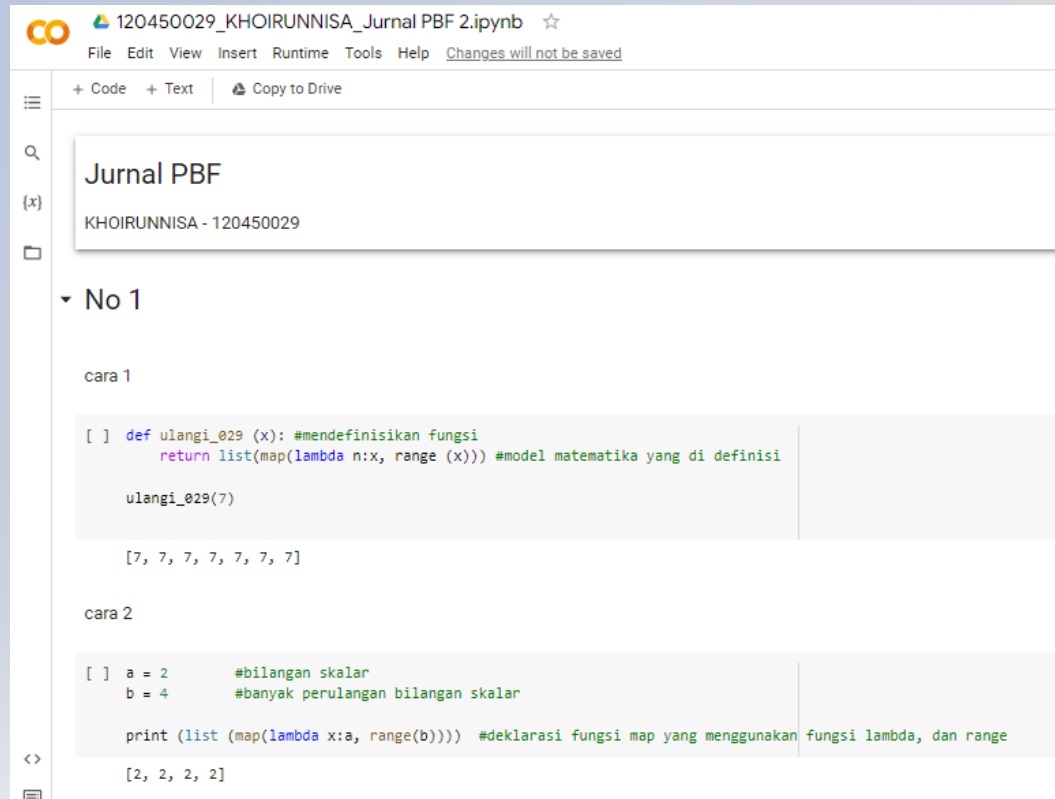
```
[ ] dna = "TGCTTCCGGCTGAGCGGTTCTTAACCAAGCAGACTGATACTGGTCGAATATCGACGGGCAAGAGCCCTGGGATTGATGCGTTTCACCATGCGCGTCTCAGTGCAGGCAGGAATGCAGAGCTTACTTC
```

```
[ ] def count_matches(dna, pattern):  
    indexes = range(0, len(dna)-len(pattern))  
    matches_start_indexes = filter(lambda start_index: dna[start_index:start_index+len(pattern)] == pattern, indexes)  
    return len(list(matches_start_indexes))
```

```
count_matches(dna, "ACT")
```

106

Jurnal 2



The screenshot shows a Jupyter Notebook interface. The title bar reads '120450029_KHOIRUNNISA_Jurnal PBF 2.ipynb'. The notebook has a sidebar with a search icon and a file icon. The main content area is titled 'Jurnal PBF' and shows a cell with the following code:

```
[ ] def ulangi_029 (x): #mendefinisikan fungsi
    return list(map(lambda n:x, range (x))) #model matematika yang di definisi

ulangan_029(7)
```

The output of the cell is:

```
[7, 7, 7, 7, 7, 7, 7]
```

Below this, there is another cell titled 'cara 2' with the following code:

```
[ ] a = 2      #bilangan skalar
    b = 4      #banyak perulangan bilangan skalar

print (list (map(lambda x:a, range(b)))) #deklarasi fungsi map yang menggunakan fungsi lambda, dan range
```

The output of this cell is:

```
[2, 2, 2, 2]
```

no 2

```
[ ] n = 7 #banyak deret
    deret = list (map(lambda x: ((-1)**(x+1)) * (1/(2**x)), range(1,n+1)))
    #
    print (deret)
```

```
[0.5, -0.25, 0.125, -0.0625, 0.03125, -0.015625, 0.0078125]
```

cara 2

```
[ ] n = int(input('masukkan nilai: ')) #declare n
    a = range(1,n+1) #declare deret a menggunakan range

def ulangi_029(x):
    return ((-1)**(x+1)) * (1/(2**x)) # mendefine model matematika fungsi ulangi_032()

print(list(map(ulangan_029, a))) # deklarasi deret dengan fungsi map()
```

```
masukkan nilai: 6
[0.5, -0.25, 0.125, -0.0625, 0.03125, -0.015625]
```

Jurnal 2

no 3

```
[ ] from functools import reduce
```

```
[ ] print (reduce(lambda x,y: x+y,deret))  
# declare jumlah deret menggunakan lambda dan built in ke dalam fungsi reduce  
  
0.3359375
```

no 4

```
[ ] data = open('DNA.txt','r')  
data = data.read()  
#fungsi untuk membaca jenis file txt/text
```

```
[ ] file = 'DNA.txt'  
dat = open(file, 'r').read() #mengupload dan open file  
dat = dat[:-1] #declare  
seq = 'ACT'  
dat #  
  
'TGTCTTCCGGCTGAGCGGTTCTAACCAGCAGACTGATACTGGTCGAATATCGACGGGCAAGAGCCCTGGGATTGATGCGTTTCACCATGCGCGTCTCAGTG
```



```
[ ] def append_n (dat, i, n):  
    return reduce(lambda a,b: a+b, dat[i:i+n])  
  
def remap(dat,seq):  
    return map(lambda x: append_n(dat,x,len(seq)), range(len(dat) - len(seq))) #define dan model fungsi remap untuk  
  
def count_mer(dat,seq):  
    return reduce(lambda a,b: a + (1 if b==seq else 0), remap(dat,seq), 0) #model dari fungsi count_mer  
  
[ ] sequences = ['A', 'AT', 'GGT', 'AAGC', 'AGCTA']  
  
def count_all(dat, sequences):  
    return map(lambda x:count_mer(dat,x), sequences)  
  
res = count_all(dat,sequences)  
print(* res) # declare ras dengan fungsi yang sudah di define  
  
2112 556 77 22 5
```



▼ No 5

```
[ ] def komplemen (x):  
    return {'A':'T','T':'A','C':'G','G':'C'}.get(x)
```

#definisi komplemen digunakan untuk mendefine reverse komplemen. sehingga pola huruf berubah dan berganti sesuai def komplemen

```
[ ] def reverse_komplemen(dat):  
    return map(lambda x:komplemen(x), dat)
```

```
[ ] res = reverse_komplemen (dat)  
print(*res)
```

A C A G A A G G C C G A C T C G C C A A G G A T T G G T C G T C T G A C T A T G A C C A G C T T A T A G C T G C C C G T T C T C



Jurnal 2

▼ No 6

```
[ ] import math #import fungsi math untuk menyelesaikan persoalan tersebut

def aktivasi(x):
    return 1/ (1+ math.exp(-x))

def WTi(W,i):
    return list(map(lambda w:w[i], W))

def WT(W):
    return list(map(lambda i : WTi(W, i), range(len(W[0]))))

def XW(X,W):
    return map( lambda w: reduce (lambda a,b:a+b, map(lambda xx, ww: xx * ww, X, w), 0), WT(W))

def input_to_hidden(X,W):
    return list(map(lambda x:aktivasi(x), XW(X,W)))

def feed_forward(X,W,M):
    return input_to_hidden(input_to_hidden(X,W), M)
```

```
[ ] X = [9, 10, -4]
W = [[0.5, 0.4], [0.3, 0.7], [0.25, 0.9]]
M = [[0.34],[0.45]]
```

```
feed_forward(X, W, M)
```

```
[0.6876336740661236]
```

```
WTi(W,0)
```

```
[0.5, 0.3, 0.25]
```

```
[ ] X
```

```
[9, 10, -4]
```



Kumpulan Tugas Mingguan (pasca UTS)

Pemrograman Berbasis Fungsi

Tugas 9

Buat program untuk menghitung deret bilangan prima dari 2 hingga N menggunakan HOF filter dan map

Contoh primes(100):

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97



```
#lecture 9
#029
def bil(n):
    return list(filter(lambda x: n % x == 0, (range (1,n+1))))
def prima(N):
    return list(filter(lambda N: len (bil (N)) == 2, range (1, N+1)))
print (prima(100))
```

[3] ✓ 0.2s

... [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

Tugas 9

```
employee = {  
    'Nagao':35,  
    'Ishii':30,  
    'Kazutomo':20,  
    'Saito':25,  
    'Hidemi':29  
}
```

Terdapat dictionary employee berisi nama dan umur pegawai, lakukan filter untuk mengetahui pegawai yang berumur > 25 tahun !



#029

```
employee = {  
    'Nagao' : 35,  
    'Ishii' : 30,  
    'Kazutomo' : 20,  
    'Saito' : 25,  
    'Hidemi' : 29  
}
```

```
print("Filter Umur: ")
```

```
print(dict(filter(lambda x:x[1] >= 25, employee.items())))
```

4] ✓ 0.6s

Filter Umur:

```
{'Nagao': 35, 'Ishii': 30, 'Saito': 25, 'Hidemi': 29}
```

Tugas 10

Latihan

1. Buat fungsi mencari jumlah bilangan genap dari list L!



```
#lecturer 10
#029
from functools import reduce

x = [2,4,1,0,5,2,9]
reduce (lambda a,b:a + (1 if b%2 == 0 else 0), x, 0)
```

[10] ✓ 0.6s

... 4

Tugas 10

Latihan

2. Buat fungsi untuk menghitung n! Menggunakan reduce!



```
#029
n = 5
print (reduce (lambda x,y: x*y, range (1,n+1)))
```

✓ 0.1s

120

Tugas 10

Latihan

3. Hitung euclidian distance dari dua vektor berikut menggunakan higher order function!

X = [2,5,6,7,10]

Y = [-2,9,2,-1,10]

#029

X = [2,5,6,7,10]

Y = [-2,9,2,-1,10]

```
euclid = lambda X,Y: reduce( lambda a,c:a+c ,map( lambda x,y: (x-y)**2 ,X, Y) )**0.5  
euclid(X,Y)
```

✓ 0.2s

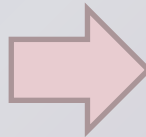
10.583005244258363

Tugas 10

Latihan

3. Terdapat dictionary employee berisi nama dan umur pegawai, lakukan reduce untuk mengetahui berapa jumlah pegawai yang berumur > 25 tahun !

```
employee = {  
    'Nagao':35,  
    'Ishii':30,  
    'Kazutomo':20,  
    'Saito':25,  
    'Hidemi':29  
}
```

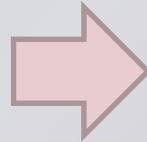


```
#029  
employee = {  
    'Nagao' : 35,  
    'Ishii' : 30,  
    'Kazutomo' : 20,  
    'Saito' : 25,  
    'Hidemi' : 29}  
print("filter by age: ")  
print(dict(filter(lambda x: x[1] >= 25,employee.items())))  
  
[14] ✓ 0.2s  
... filter by age:  
{'Nagao': 35, 'Ishii': 30, 'Saito': 25, 'Hidemi': 29}
```

Tugas 10

Latihan

4. Buatlah deret fibonacci menggunakan higher order function!



```
... Fibonacci 0 -> [0]
Fibonacci 1 -> [0, 1]
Fibonacci 2 -> [0, 1, 1]
Fibonacci 3 -> [0, 1, 1, 2]
Fibonacci 4 -> [0, 1, 1, 2, 3]
Fibonacci 5 -> [0, 1, 1, 2, 3, 5]
Fibonacci 6 -> [0, 1, 1, 2, 3, 5, 8]
Fibonacci 7 -> [0, 1, 1, 2, 3, 5, 8, 13]
Fibonacci 8 -> [0, 1, 1, 2, 3, 5, 8, 13, 21]
Fibonacci 9 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
Fibonacci 10 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
Fibonacci 11 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
Fibonacci 12 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144]
Fibonacci 13 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233]
Fibonacci 14 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377]
Fibonacci 15 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]
Fibonacci 16 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987]
Fibonacci 17 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597]
Fibonacci 18 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584]
Fibonacci 19 -> [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181]

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]
```

#029

```
fibo = lambda n: reduce( lambda a,b: a if b[0] <=1 else a + [a [ b[0]-1] + a[b[0]-2] ],enumerate ( [0,1] + list( range(1,n) ) ), [0,1] ) if n > 0 else [0]

for i in range(20):
    print('Fibonacci',i,'->', fibo(i))
fibo(15)
```

✓ 0.1s

Pyth

Tugas 11

Latihan

Buat sebuah program untuk membuat deret fibonacci dari 0 hingga N dengan menggunakan fungsi non-rekursif dan rekursif!

Bandingkan keduanya jika nilai N = 500, Manakah yang lebih baik? Jelaskan!

```
#Lecture 11
#Fungsi Recursive
```

```
fibonacci = lambda n: 0 if n==0 else 1 if ( n==1 or n==2) else fibonacci(n-1) + fibonacci(n-2)
deret_fibonacci = lambda n: list( map( lambda i : fibonacci(i), range(n+1)))
```

```
deret_fibonacci(500)
```

KeyboardInterrupt Traceback (most recent call last)

```
c:\Users\ACER\Downloads\Untitled-1.ipynb Cell 17' in <cell line: 7>()
    4 fibonacci = lambda n: 0 if n==0 else 1 if ( n==1 or n==2) else fibonacci(n-1) + fibonacci(n-2)
    5 deret_fibonacci = lambda n: list( map( lambda i : fibonacci(i), range(n+1)))
----> 7 deret_fibonacci(500)
```

```
c:\Users\ACER\Downloads\Untitled-1.ipynb Cell 17' in <lambda>(n)
    1 #Lecture 11
    2 #Fungsi Recursive
    4 fibonacci = lambda n: 0 if n==0 else 1 if ( n==1 or n==2) else fibonacci(n-1) + fibonacci(n-2)
----> 5 deret_fibonacci = lambda n: list( map( lambda i : fibonacci(i), range(n+1)))
    7 deret_fibonacci(500)
```

```
c:\Users\ACER\Downloads\Untitled-1.ipynb Cell 17' in <lambda>(i)
    1 #Lecture 11
    2 #Fungsi Recursive
    4 fibonacci = lambda n: 0 if n==0 else 1 if ( n==1 or n==2) else fibonacci(n-1) + fibonacci(n-2)
----> 5 deret_fibonacci = lambda n: list( map( lambda i : fibonacci(i), range(n+1)))
```

```
#Non Recursive menggunakan Reduce
from functools import reduce as r

fibonacci = lambda n: r( lambda a,b: a if b[0] <=1 else a + [a [ b[0]-1] + a[b[0]-2] ],
                        enumerate ( [0,1] + list( range(1,n) ) ), [0,1] ) if n > 0 else [0]

fibonacci(500)
```

[5] ✓ 0.3s

... Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```
[0,
1,
1,
2,
3,
5,
8,
13,
21,
34,
55,
89,
144,
233,
377,
610,
987
```

Tugas 11

Jadi kesimpulan dari percobaan Lecture 11 dalam membuat fibonacci, lebih baik menggunakan non recursive terlihat dari hasil running yang lebih cepat

Tugas 12

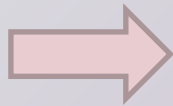
Latihan 1

Ubah fungsiku menjadi pure function!

```
[12]: def fungsiku(L):  
      def check_genap(l):  
          return l % 2 == 0  
      for i in range(len(L)):  
          if check_genap(L[i]):  
              L[i] = L[i]/2  
          else:  
              L[i] = L[i] * n + 1  
      return L
```

```
[13]: n = 3  
      L = [5,6,7,8]  
      print(fungsiku(L))  
      [16, 3.0, 22, 4.0]
```

```
[14]: print(L)  
      [16, 3.0, 22, 4.0]
```



```
#lecturer 12  
  
n = 3  
L = [5,6,7,8]  
def fungsiku(L, n):  
    return list(map(lambda x: x/2 if x%2==0 else x*n+1, L))  
print(fungsiku(L,n))  
print(L)  
  
[7] ✓ 0.2s Python  
... [16, 3.0, 22, 4.0]  
     [5, 6, 7, 8]
```

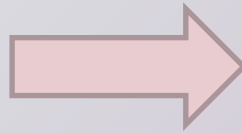

Tugas 12

Latihan 2

Ubah fungsiku2 menjadi pure function!

```
[19]: def fungsiku2(L):  
      def check_faktor(l):  
          return l % n == 0  
      for i in range(len(L)):  
          if check_faktor(L[i]):  
              L[i] = L[i]/2  
          else:  
              L[i] = L[i] * n + 1  
      return L
```

```
[21]: n = 3  
      L = [5,6,7,8]  
      print(list(fungsiku2(L)))  
      print(L)  
  
      [16, 3.0, 22, 25]  
      [16, 3.0, 22, 25]
```



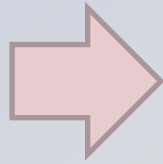
```
▶ n = 3  
  L = [5,6,7,8]  
  def fungsiku2(L, n):  
      return list(map(lambda x: x/2 if x%n==0 else x*n+1, L))  
  print(fungsiku2(L,n))  
  print(L)  
[10] ✓ 0.2s Python  
... [16, 3.0, 22, 25]  
     [5, 6, 7, 8]
```

Tugas 12

Latihan 3

Apakah isi dalam tupel tup ada yang dapat diubah?

```
tup = ([3, 4, 5], 'myname')
```



```
tup = ( [3,4,5], 'kirun' )
tup[0].append(6)
print(tup)

# dapat diubah jika terdapat mutable di dalam tuple
```

[11] ✓ 0.1s Python

```
... ([3, 4, 5, 6], 'kirun')
```

Tugas 13

Latihan

Buatlah fungsi untuk menghitung biaya yang harus dibayar customer pada suatu e-commerce menggunakan higher order function. Buatlah decorator untuk mengeluarkan harga sebelum pajak dan sesudah pajak (pajak = 11%) ! Gunakan decorator untuk menambahkan perhitungan waktu eksekusi!

Contoh input:

```
keranjang = [  
    {'Jumlah_Barang': 5 , 'Harga': 10 },  
    {'Jumlah_Barang': 7 , 'Harga': 20 },  
    {'Jumlah_Barang': 20 , 'Harga': 4.5 }  
]
```



Tugas 13

```
from functools import reduce as r
keranjang = [
    {'Jumlah_Barang': 5, 'Harga': 10},
    {'Jumlah_Barang': 7, 'Harga': 20},
    {'Jumlah_Barang': 20, 'Harga': 4.5}
]
def pajak_decorator(func):
    def inner(*args, **kwargs):
        res = func(*args, **kwargs)
        print('Sub Total: ', res)
        print('Pajak: ', res * 0.11)
        print('Total: ', res + res * 0.11)
        return res
    return inner
import time
def calc_time_decorator(fu):
    def inner(*args, **kwargs):
        waktu_awal = time.time()
        res = fu(*args, **kwargs)
        waktu_akhir = time.time()
        print('Waktu eksekusi: ', waktu_akhir - waktu_awal)
        return res
    return inner
```

✓ 0.7s

Python

```
@calc_time_decorator
@pajak_decorator
def hitung_pembayaran_1(keranjang):
    return r(lambda a,b: a + (b['Jumlah_Barang'] * b['Harga']), keranjang, 0)
hitung_pembayaran_1(keranjang)
```

19] ✓ 0.1s

```
.. Sub Total: 280.0
   Pajak: 30.8
   Total: 310.8
   Waktu eksekusi: 0.00501704216003418
```

280.0

```
@calc_time_decorator
@pajak_decorator
def hitung_pembayaran_2(keranjang):
    s = 0
    for k in keranjang:
        s = s + k['Jumlah_Barang'] * k['Harga']
    return s * 1000
hitung_pembayaran_2(keranjang)
```

1] ✓ 0.1s

```
· Sub Total: 280000.0
  Pajak: 30800.0
  Total: 310800.0
  Waktu eksekusi: 0.002512216567993164
```

280000.0

Tugas 13

Latihan

Addku = lambda x: x + 10

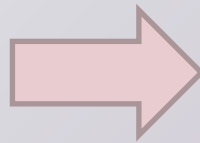
Powku = lambda x: x**2

Kurku = lambda x: x - 2 * x

a. Buatlah fungsi komposisi menggunakan 3 fungsi diatas yang melakukan hal sebagai berikut secara berurut:

1. Menjumlahkan input dengan nilai 10
2. Mengurangi input dengan 2 kali input nya
3. Mengeluarkan nilai kuadrat dari input nya

B. Buatlah fungsi invers nya!



#lecture 13

```
addku = lambda x:x+10
```

```
powku = lambda x: x**2
```

```
kurku = lambda x:x-2*x
```

```
f_komp = lambda f,g: lambda x: f(g(x))
```

```
my_f_kom = f_komp(kurku ,f_komp(powku, addku))
```

```
my_f_kom(10)
```

[12] ✓ 0.1s

Python

... -400

▷

invers

```
inv_addku=lambda x:x-10
```

```
inv_powku=lambda x: x**8.
```

```
inv_kurku=lambda x: -1 *x
```

```
my_f_kom_inv = f_komp( inv_addku, f_komp(inv_powku, inv_kurku))
```

```
my_f_kom_inv(-400)
```

[13] ✓ 0.1s

Python

... 6.5536e+20

Tugas 13



Latihan Penentuan UKT Mahasiswa

Universitas di Lampung ITARE, ingin memiliki sistem penentuan golongan UKT dan jumlah biaya UKT yang dibayarkan oleh Mahasiswa berdasarkan Kriteria berikut:

1. Jumlah tanggungan
2. Jumlah token listrik selama 3 bulan terakhir
3. Gaji Orang tua / Penanggung jawab
4. Penerima program KIP-K atau bukan

Latihan : Penentuan UKT Mahasiswa

3. Ketentuan Gaji :

Jika gaji penanggung jawab > 10 juta maka skor = 7

Jika $8 < \text{gaji} \leq 10$ juta, maka skor = 6

Jika $6 < \text{gaji} \leq 8$ juta, maka skor = 5

Jika $4 < \text{gaji} \leq 6$ juta, maka skor = 4

Jika $3 < \text{gaji} \leq 4$ juta, maka skor = 3

Jika gaji < 3 juta, maka skor = 2

4. Jika mahasiswa memiliki KIP-K, maka skor = 1, jika tidak maka skor = 5

1. Ketentuan Jumlah Tanggungan :

Jika lebih ≥ 5 , maka skor = 1

Jika < 5, maka skor = 5 - jumlah tanggungan

2. Ketentuan token listrik:

Jika rata-rata lebih dari 100 ribu per bulan, maka skor = 3

Jika diantara 50 ribu - 100 ribu per bulan, maka skor = 2

Jika dibawah 50 ribu, maka skor = 1

Latihan : Penentuan UKT Mahasiswa

Perhitungan pembayaran UKT adalah sebagai berikut:

$$\text{Skor_total} = 20 \% * \text{skor_1} + 30 \% * \text{skor_2} + 20\% \text{ skor_3} + 30\% \text{ skor_4}$$

$$\text{Jumlah bayar UKT} = \text{biaya pokok} + \text{skor_total} * 500 \text{ ribu}$$

$$\text{Uang Pokok} = 750 \text{ ribu}$$

Latihan : Penentuan UKT Mahasiswa

Gunakan fungsi komposisi untuk menyelesaikan masalah tersebut!

Hitung berapa biaya UKT yang harus dibayarkan dengan input sebagai berikut:

1. Jumlah tanggungan = 3
2. Listrik 3 bulan terakhir = 120 ribu, 75 ribu, 50 ribu
3. Gaji Penanggung jawab = 5.5 juta per bulan
4. Peserta KIP-K = Tidak



Tugas 13



```
from functools import reduce as r

#Define function composition
mycompose = lambda *funcs: r(lambda f,g:lambda x:f(g(x)), reversed(funcs), lambda x:x)

# ketentuan jumlah tanggungan
def skor1(jtg):
    ...return 1 if jtg >= 5 else 5-jtg

# ketentuan token listrik
from functools import reduce as r

def skor2(X):
    ...def rata(X):
    ...    ...return sum(X) / len(X)
    ...
    ...def l_cond_1(X):
    ...    ...return [X, [X>100000]]
    ...
    ...def l_cond_2(X):
    ...    ...return [x[0], x[1] + [x[0] >= 50000]]
    ...
    ...def to_score2(X):
    ...    ...return r(lambda a,b: a + (1 if b == True else 0), X[1], 1)
    ...
    ...compose_cond = mycompose(rata, l_cond_1, l_cond_2, to_score2)
    ...return compose_cond(X)
```

Tugas 13

```
# ketentuan gaji

def con_1(X):
    ... return [X[0], 1, X[2], [X[0] > X[2][X[1]]]]
def con_2_to_n(X):
    ... return [X[0], X[1]+1, X[2], X[3]+[X[0] > X[2][X[1]]]]
def to_score(X):
    ... return r(lambda a,b: a+(1 if b==True else 0), X[-1],2)
def prep(gj):
    ... return [gj, 0, list(map(lambda x: x*1800000, list(range(10,3,-2)) + [3]))]
def skor3(gaji):
    ... compy = mycompose(prep, con_1, *(con_2_to_n for i in range(4)), to_score)
    ... return compy(gaji)
    .....
```

```
# ketentuan KIP-k
def skor4(X=True):
    ... return 1 if X else 5
```

```
# ketentuan KIP-k
def skor4(X=True):
    ... return 1 if X else 5
```

[12] ✓ 0.1s

```
from functools import reduce as r

def combineskor(X):
    return X + [map(lambda f,x: f(x), X[1], X[0])]
def boboti(X):
    return r(lambda a,b:a+b, map(lambda x,y:x*y, X[-1], [0.2, 0.3, 0.2, 0.3]))
def toUKT(X):
    return 750000 + X*500000
```

[13] ✓ 0.1s

```
mhs =[3, [120000, 75000, 50000], 5.5 * 10**6, False ]

datas = [ mhs, [skor1, skor2, skor3, skor4] ]
compose_fin = mycompose(combineskor, boboti, toUKT)
jumlah = compose_fin(datas) / 10**6
print ("Jumlah UKT: ", jumlah)
```

[21] ✓ 0.8s

... Jumlah UKT: 2.3



Tugas 13

Latihan: Turunan Polinom

Contoh input:

```
dat = '-3x^5 + 2x^2 - 4x + 5'
```

Output:

```
' -15.0x^4 + 4.0x -4.0'
```

```
#029
#Turunan polinom

def split(dat):
    return dat.replace(' ', '').replace('-', '+-').split('+')

def chdepan(dat):
    return dat[1:] if dat[0] == '-' else dat

def eqkan(dat):
    return map( lambda x: x if '^' in x else x+ '^1' if 'x' in x else x+ 'x^0', dat)

def toarr2d(dat):
    return r( lambda a, b: a + [[float(hurf) for hurf in b.split('x^')]] , dat, [])

def sortdesc(dat):
    return sorted(dat, key=lambda x: x[1], reverse=True)

def calctur(dat):
    return map( lambda x: [0,0] if x[1] == 0 else [x[1]*x[0], x[1]-1], dat)

def tostr(dat):
    return map( lambda x: '0' if x[0] == 0 else str(x[0]) if x[1]==0 else str(x[0]) + 'x^' + str(x[1]), dat)

def prettykan(dat):
    return r( lambda a,b: a+'+' + b if b != '0' else a, dat, '')

def prettysign(dat):
    return dat.replace('+-', ' -').replace('+', ' +')
```

```
✓ [17] compose_lambda = lambda *funcs: r(lambda f,g:lambda x:f(g(x)), reversed(funcs), lambda x:x)
0s mycomp_lambda = compose_lambda(addku,powku,kurku)
```

```
✓ [14] dat = '-3x^5 + 2x^2 -4x +5'
0s fss = (split, chdepan, eqkan, toarr2d, sortdesc, calctur, tostr, prettykan, prettysign)
my_turunan = compose_lambda(*fss)
my_turunan(dat)

' -15.0x^4.0+ 4.0x^1.0 -4.0'
```

thank you!