

2022

Sains Data

Intitut Teknologi Sumatera

Laporan Akhir Praktikum

Pemrograman Berbasis Fungsi



Akbar Fadhillah Irsyad Usman

120450026

||



JURNAL

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!

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


```
kategoris = {  
    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  
  
def kabisat(thn):  
    return ( is_awal_abad(thn) and thn % 400 == 0 ) or ( not is_awal_abad(thn) and thn % 4 == 0 )  
  
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 kabisat(t[0]) else 28 )  
  
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(tgl_p,d,dec_c(tgl_p)) ]
```

▶ # 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]) * durasi
```

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

62500

JURNAL 2

1.1 Soal

Kerjakan seluruh soal berikut dengan menggunakan higher order function map, filter dan reduce!

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

```
[15] from functools import reduce #untuk import fungsi reduce () sehingga dapat beroperasi
def ulangi_026(x,n): #mendefine fungsi
    return reduce(lambda a,b:a + [x] , range (n),[]) #model matematika perulangan yg di define kedalam fungsi menggunakan fungsi reduce
```

```
ulangi_026(1,10)
```

```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

```
[16] a = 1 #bil. skalar
n = 10 #banyaknya deret bilangan skalar

print(list(map(lambda x:a, range(n)))) #deklarasi fungsi map yg menggunakan fungsi lambda dan build in
```

```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

jurnal 2 no 2

2. Buatlah deret bilangan sebagai berikut dengan input n sebagai panjang deret: $1/2, -1/4, 1/8, \dots, (-1)^n \frac{1}{2^{n+1}}$

```
[ ] 1 n = 8 #declare n
    2 deret = list(map(lambda x: ((-1)**(x+1)) * (1/(2**(x+1))),range(1,n+1))) #declare deret menggunakan range
    3 print (deret) #print deret
```

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

```
[ ] 1 b = range(1,n+1) #declare deret b menggunakan range
    2
    3 def pola_deret(x): #fungsi pola pada deret
    4 | | return ((-1)**(x)) * (1/(2**(x+1)))
    5
    6 print(list(map(pola_deret, b))) #deklarasi fungsi deret dengan map
```

```
[-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!

```
[ ] 1 from functools import reduce #untuk import fungsi reduce () sehingga dapat beroperasi

[ ] 1 print(reduce(lambda x,y: x+y, deret, 0)) #fungsi declare jumlah deret menggunakan lambda dan build in kedalam reduce

0.16650390625
```

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

<https://drive.google.com/file/d/18C1ylsTXrY9pglqglhijoS8LYmcxdIjM/view?usp=sharing>

hitunglah jumlah kemunculan pola berikut pada data tersebut:

- a. A
- b. AT
- c. GGT
- d. AAGC
- e. AGCTA

```
[ ] 1 filename = 'DNA.txt' #nama file untuk mengimport data
2 dat_026 = open(filename, 'r').read() #fungsi untuk membuka/membaca file
3 dat_026 = dat_026[:-1] #declare sec
4 sec = 'ACT'
5 dat_026 #run declare dat
```

```
[ ] 1 def append_n(dat_026, i, n): #define fungsi append n
2 | | return reduce(lambda a,b:a+b , dat_026[i:i+n] ) #model fungsi append
3 def remap(ddat_026, seq): #define fungsi remap
4 | | return map(lambda x: append_n(dat_026,x , len(seq) , range(len(dat_026) - len(seq) ) ) #model fungsi remap
5 def count_mer(dat_026, seq): #define fungsi count mer
6 | | return reduce(lambda a,b: a + (1 if b==seq else 0) , remap(dat_026,seq), 0 ) #model fungsi count mer
```

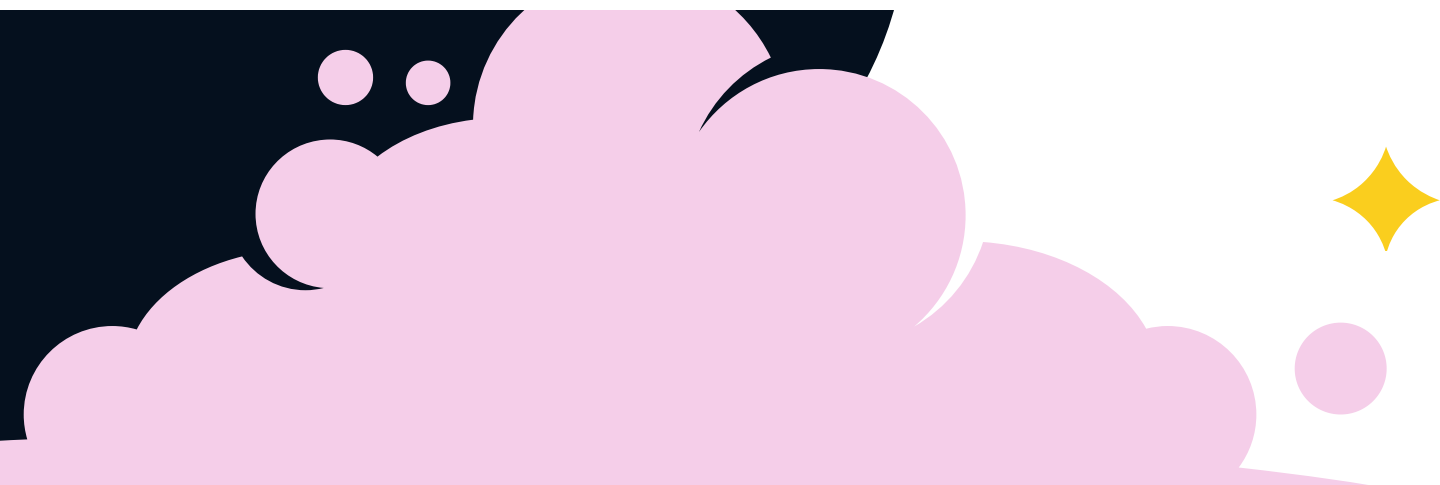
```
[ ] 1 append_n(dat_026,1,3)
```

```
[ ] 1 list(remap(dat_026, 'ACT')) #list dari fungsi remap dan dat
2 len(dat_026)
```

6930

```
[ ] 1 sequences = ['A','AT','GGT','AAGC','AGCTA'] #declare seq dari pola yang akan dicari
2
3 def count_all(dat_026,sequences): #define fungsi count all
4 | | return map(lambda x: count_mer(dat_026,x), sequences) #model fungsi count all
5
6 res = count_all(dat_026,sequences) #deklarasi res dengan fungsi yang sudah di define
7 print(* res) #menjalankan fungsi res
```

2112 556 77 22 5





5. Reverse complement dari suatu sequence string DNA memiliki aturan sebagai berikut:

A adalah komplemen dari T

C adalah komplemen dari G

Contoh reverse complement:

input DNA : ACTGA

Reverse complenet : TGACT

Buatlah fungsi untuk mencari inverse komplemen dari data pada nomor 4 !

```
[ ] 1 def komplemen(x):  
    2 | | return {'A':'T', 'T':'A', 'C':'G', 'G':'C'}.get(x)  
    3 #define komplemen yg digunakan untuk reverse komplemen  
    4  
    5 def reverse_komplemen(dat_026):  
    6 | | return map(lambda x:komplemen(x), dat_026)  
    7 #define fungsi reverse komplemen
```

```
[ ] 1 res = reverse_komplemen(dat_026)  
    2 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 G G G A C C C T A A C T A C G C A A A G T G G T A C G



6. NEURAL NETWORK

```
[ ] 1 import math #mengimport model matematika agar dapat di gunakan
2
3 def aktivasi(x):
4 | | return 1/ (1+ math.exp(-x))
5 #define fungsi aktivasi beserta modelnya
6
7 def WTi(W,i):
8 | | return list(map(lambda w:w[i], W))
9 #define fungsi WTi beserta modelnya
10
11 def WT(W):
12 | | return list(map(lambda i : WTi(W, i), range(len(W[0]))))
13 #define fungsi WT beserta modelnya
14
15 def XW(X,W):
16 | | return map(lambda w: reduce (lambda a,b:a+b, map(lambda xx, ww: xx * ww, X, w), 0), WT(W))
17 #define fungsi XW beserta modelnya
18
19 def input_to_hidden(X,W):
20 | | return list(map(lambda x:aktivasi(x), XW(X,W)))
21 #define fungsi input to hidden beserta model nya
22
23 def feed_forward(X,W,M):
24 | | return input_to_hidden(input_to_hidden(X,W), M)
25 #define fungsi input feed forward beserta modelnya
```



```
[ ] 1 X = [9, 10, -4] #declare X
2 W = [[0.5, 0.4], [0.3, 0.7], [0.25, 0.9]] #declare w
3 M = [[0.34], [0.45]] #declare M
4
5 feed_forward(X,W,M) #untuk menjalankan fungsi feed forward

[0.6876336740661236]
```

```
[ ] 1 WTi(W,0) #menjalankan WTi

[0.5, 0.3, 0.25]
```

```
[ ] 1 X # Menjalankan X

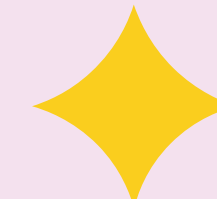
[9, 10, -4]
```

LATIHAN



Jupiter

LATIHAN 9



```
[ ] 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))
```

```
[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]
```

```
[ ] employee = {  
    'Nagao' : 35,  
    'Ishii' : 30,  
    'Kazutomo' : 20,  
    'Saito' : 25,  
    'Hidemi' : 29  
}  
  
print("Filter Umur: ")  
print(dict(filter(lambda x:x[1] >= 25, employee.items())))
```

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

```
[ ] negara = ['INA', 'Kor', 'Rus', 'Chn', 'Jpn']  
print (*filter(lambda x:x == 'INA', negara))  
print (*filter(lambda x:x != 'INA', negara))  
  
isIna = lambda x:x=='INA'  
print (*filter(isIna,negara))  
print (*filter(lambda x:not isIna(x),negara))
```

```
INA Kor Rus Chn Jpn  
INA  
Kor Rus Chn Jpn
```

```
[ ] L = [2,False, True, 2,5,7]  
print (*filter(None,L))
```

```
2 True 2 5 7
```

LATIHAN 10

```
[ ] 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)
```

4

```
[ ] n = 5
```

```
print (reduce (lambda x,y: x*y, range (1,n+1)))
```

120

```
[ ] 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)
```

10.583005244258363

```
[ ] 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())))
```

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

```
fibonacci = 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,'->', fibonacci(i))
fibonacci(15)
```

```
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]
```

LATIHAN 11

↳ latihan 11

```
[ ] # recursive
```

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

deret_fibonacci_rec(10)
```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

```
[ ] #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(10)
```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

LATIHAN 12

latihan 12

```
[ ] def fungsiku(cL):  
    L = cL.copy()  
    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
```

```
[ ] n = 3  
L = [5, 6, 7, 8]  
print(fungsiku(L))
```

```
[16, 3.0, 22, 4.0]
```

```
[ ] print(L)
```

```
[5, 6, 7, 8]
```

```
[ ] 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)
```

```
[16, 3.0, 22, 4.0]  
[5, 6, 7, 8]
```

latihan 2

ubah fungsiku2 menjadi pure function!

```
[ ] 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
```

```
[ ] 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)
```

```
[16, 3.0, 22, 25]  
[5, 6, 7, 8]
```

latihan 3

Apakah isi dalam tupel tup ada yang dapat diubah?

jawaban : dapat diubah jika terdapat mutable di dalam tuple

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

```
([3, 4, 5, 6], 'akbar')
```

Latihan 13

```
[ ] from functools import reduce as r
```

```
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)
```

-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)
```

6.5536e+20

```
[ ] @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)
```

```
Sub Total: 280000.0
Pajak: 30800.0
Total: 310800.0
Waktu eksekusi: 0.003570079803466797
280000.0
```

```
[ ] #029
#Turnan 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('+', ' + ')
```

```
[ ] 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
```

```
@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)
```



```
[ ] @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)
```

```
Sub Total: 280000.0
Pajak: 30000.0
Total: 310000.0
Waktu eksekusi: 0.003570079803466797
280000.0
```

```
[ ] #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('+', '+ ')
```

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

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

Terima Kasih

Jupiter