

**KUMPULAN TUGAS PEMROGRAMAN BERBASIS FUNGSI**



**DEVRI ZEFFANYA**

**: 120450105**

**KELAS RA**

**INSTITUT TEKNOLOGI SUMATERA**

**LAMPUNG SELATAN**

**2022**

## **JURNAL PRAKTIKUM MODUL 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 penye-

waan 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!

### **Jawaban**

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 penye-

waan 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 (hari): '))
```

```
kategori = {  
    1: 100,  
    2: 200,  
    3: 250,  
    4: 300,  
    5: 500,  
}
```

```
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))]

tgl_p = dtl(tanggal)
wkt_kembali(tgl_p,durasi)

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)

```

## JURNAL MODUL 2

### Soal

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

1. Buatlah sebuah fungsi bernama ulang\_NIM, ulang memiliki input sebuah bilangan skalar a, dan mengeluarkan vektor 1xn dengan seluruh elemen nya adalah a !
2. Buatlah deret bilangan sebagai berikut dengan input n sebagai panjang deret:

$$\frac{1}{2}, -\frac{1}{4}, \frac{1}{8}, \dots, (-1)^n \frac{1}{2^{n+1}}$$

3. Jumlahkan deret bilangan tersebut!
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>

hitunglah jumlah kemunculan pola berikut pada data tersebut:

- a. A
  - b. AT
  - c. GGT
  - d. AAGC
  - e. AGCTA
5. Reverse complement dari suatu sequence string DNA memiliki aturan sebagai berikut:

A adalah komplemen dari T

C adalah komplemen dari G

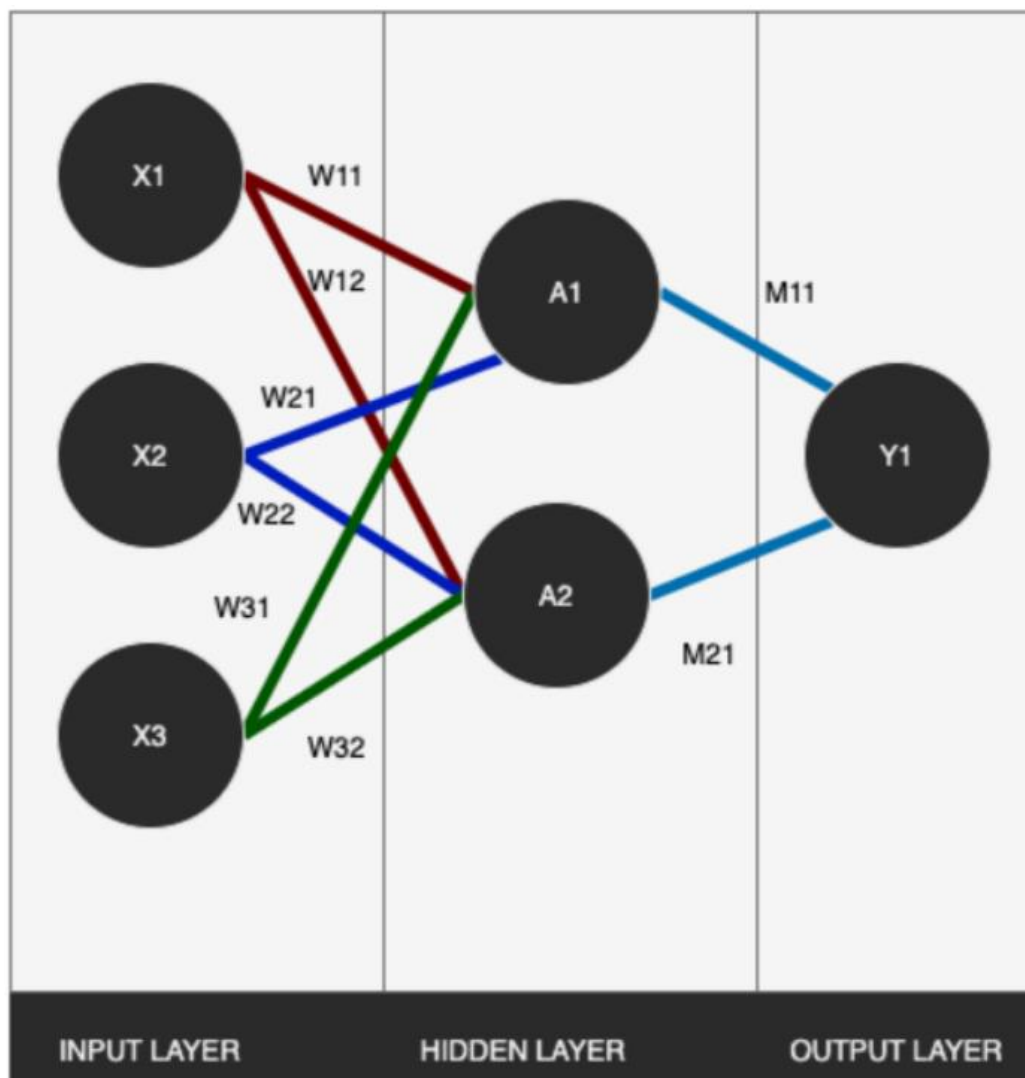
Contoh reverse complement:

input DNA : ACTGA

Reverse complement : TGACT

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

6. Perhatikan Neural Network dibawah ini:



Terdapat proses yang dinamakan feed-forward. Input dalam sebuah neural network diproses ke hidden layer hingga ke output layer.

Setiap Node, menunjukan neuron dan setiap garis menunjukan weight.

Proses Feed-Forward berjalan dari input layer menuju output layer.

Nilai yang masuk ke neuron di hidden layer adalah penjumlahan antara perkalian weight dengan nilai yang masuk pada input neuron setelah itu diaktifkan dengan fungsi aktivasi. Atau dapat dimodelkan sebagai berikut:

$$S_1 = X_1.W_{11} + X_2.W_{21} + X_3.W_{31}$$

$$S_2 = X_1.W_{12} + X_2.W_{22} + X_3.W_{32}$$

$$A_1 = \frac{1}{1 + e^{-S_1}}$$

$$A_2 = \frac{1}{1 + e^{-S_2}}$$

$$Z_1 = M_{11}.A_1 + M_{21}.A_2$$

$$Y_1 = \frac{1}{1 + e^{-Z_1}}$$

Buatlah fungsi feed-forward dengan input berikut:

W11 = 0.5

W12 = 0.4

W21 = 0.3

W22 = 0.7

W31 = 0.25

W32 = 0.9

M11 = 0.34

M21 = 0.45

dan X1 = 9 , X2 = 10 , X3 = -4

**Jawaban**

```
"""1."""
```

```
a = int(input("masukkan jumlah bilangan : "))
l = list(map(int,input("masukkan bilangan : ").strip().split()))
def ulangi_105(n):
    return n * 1
print(list(map(ulangi_105, l)))
```

```
"""2."""
```

```
n = 10
print(list(map(lambda x: ((-1)**(x+1))*(1/(2**x)), range(1,n))))
```

```
"""3."""
```

```
l = [0.5, -0.25, 0.125, -0.0625, 0.03125, -0.015625, 0.0078125, -
0.00390625, 0.001953125]
def add(a,b):
    deret_bil = a+b
    print('a:',a,', b:',b, '-> a+b:',deret_bil)
    return deret_bil
print(( add, l ))
```

```
"""4. """
```

```
dna =
"TGTCCTTCCGGCTGAGCGGTTCCCTAACAGCAGACTGATACTGGTCGAATATCGACGGGCAAGAGCCCTGG
GATTGATGCGTTT"
def count_matches(dna, pattern):
    indexes = range(0, len(dna)-len(pattern))
    callback = lambda start_index:
dna[start_index:start_index+len(pattern)] == pattern
    matches_start_indexes = filter(callback, indexes)
    return len(list(matches_start_indexes))
print("A: {}".format(count_matches(dna, "A")))
print("AT: {}".format(count_matches(dna, "AT")))
print("GGT: {}".format(count_matches(dna, "GGT")))
print("AAGC: {}".format(count_matches(dna, "AAGC")))
print("AGCTA: {}".format(count_matches(dna, "AGCTA")))
```

```
"""5."""
```

```
def complement(g):
    return {"A": "T", "C": "G", "T": "A", "G": "C" }.get(g)
def reverse_complement(dna):
    return "".join(map(complement, list(dna)))
reverse_complement("ACTGA")
```

```
"""6. """
```

```
import math
```

```
layer_0_1 = [
[0.5 , 0.4],
[0.3, 0.7],
[0.25, 0.9]
]
layer_1_2 = [ [0.34], [0.45]]
x = [[9, 10, -4]]
```

```
def sigmoid(x):
    return 1 / (1 + math.exp(-1 * x))
```

```
def multiply(x, y):
    return map(lambda z: z[0] * z[1], zip(x, y))
```

```
def reduce_sum(arr):
    return functools.reduce(lambda acc, cur : acc+cur, arr, 0)
```

```
def map_2d(fn, a):
    return list(map(lambda xi: list(map(lambda x: fn(x), xi)), a))
```

```

def transpose(x):
    return list(map(lambda row: list(map(lambda col: x[col][row],
range(len(x)))), range(len)

def dot(a, b):
    return list(map(lambda ai: list(map(lambda ti:
reduce_sum(multiply(ai, ti)), transpose(b)

def weighted_sum(inp, out):
    return lambda afn: map_2d(afn, dot(inp, out))

def feedforward(input_layer, hidden_layers, output_layer,
activation_function):

layers = [*hidden_layers, output_layer]
reducer = lambda prev, next : weighted_sum(prev,
next)(activation_function)
    return functools.reduce(reducer, layers, input_layer)
feedforward(X, [layer_0_1], layer_1_2, sigmoid)

```

## LATIHAN SOAL 9

### Soal

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

```

2. 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 !

### Jawaban

```

1.
"""

L = lambda n: range(2,n)
faktor = lambda n: list(filter(lambda i: n%i==0, L(n)))
primes = lambda n: filter (lambda i: len(faktor(i))==0, range(2,n+1))
print(*primes(100))

```



```

"""2."""

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

print(list(employee.items()))
print("Saring dari umur: ",* filter (lambda X:X[1]> 25,
employee.items()))

```

## LATIHAN SOAL 10

### Soal

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

Contoh:

L = [2,1,9,10,3,90,15]

Output: 3

2. Buat fungsi untuk menghitung n! Menggunakan reduce!

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

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

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

4. Buatlah deret fibonacci menggunakan higher order function!

### Jawaban

```

1.
"""

L = [2,1,9,10,3,90,15]
(lambda a,b:a+(1 if b % 2 == 0 else 0), L,0)

"""2."""

from functools import reduce

n = 7
print(reduce(lambda x,y: x*y, range (1,n+1)))

"""3. """

X = [2, 5, 6, 7, 10]
Y = [-2, 9, 2, -1, 10]
euclidian = lambda X,Y: r(lambda a,c:a+c, map(lambda x,y: (x-y)**2,
X,Y))**0.5

```

```

euclidian(X,Y)

"""3. """

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

cnt_emp = lambda lin, employee: r( lambda a,b: a+1 if b[1]>lin else a,
employee.items(),0)
cnt_emp(25,employee)

"""4. """

fibo = 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]
for i in range(10):
    print('Fibonachi', i,':',fibo(i))

```

## LATIHAN SOAL 11

### Soal

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!

### Jawaban

## LATIHAN SOAL 12

### Soal

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]
```

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]
```

3. Apakah isi dalam tuple tup ada yang dapat diubah?

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

## Jawaban

### LATIHAN SOAL 13

#### Soal

1. 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!

2. 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

1. Ketentuan Jumlah Tanggungan :

Jika lebih  $\geq 5$ , 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

### 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

Perhitungan pembayaran UKT adalah sebagai berikut:

$\text{Skor\_total} = 20 \% * \text{skor\_1} + 30 \% * \text{skor\_2} + 20\% \text{ skor\_3} + 30\% \text{ skor\_4}$

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

Uang Pokok = 750 ribu

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

3. 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 }  
]
```

**Jawaban**