VERSION 0.1

SEPTEMBER , 2023

[PRAKTIKUM PEMROG. FUNGSIONAL]

MODUL UAP – UJIAN AKHIR PRAKTIKUM

DISUSUN OLEH :
Fildzah Lathifah
Hania Pratiwi Ningrum

DIAUDIT OLEH
Fera Putri Ayu L., S.Kom., M.T.

PRESENTED BY: TIM LAB-IT UNIVERSITAS MUHAMMADIYAH MALANG

PERSIAPAN MATERI

Praktikan diharapkan sudah mempelajari materi dari modul 1 – 6.

TUJUAN PRAKTIKUM

1. Praktikan diharap dapat menyelesaikan semua soal yang diberikan

PERSIAPAN SOFTWARE/APLIKASI

- Komputer/Laptop
- Sistem operasi Windows/Linux/Mac OS/Android

MATERI POKOK

Pada pemrograman fungsional, modul juga dapat diakses melalui google collab agar lebih interaktif :

Modul UAP

Soal Ujian Akhir Praktikum

Pemrograman Fungsional

1. Sequence

1. Buatlah sebuah tuple yang berisi nama-nama barang beserta harga nya seperti berikut: Pensil, 1500, Buku, 5000, Penggaris, 2000.
2. Buatlah sebuah fungsi untuk memisahkan data pada no.1 tersebut menjadi 2 list berdasarkan tipe datanya.
3. Buatlah sebuah fungsi untuk menggabungkan kedua list pada hasil no 2 menjadi sebuah dictionary berisi pasangan nama barang dan harganya.
4. Tampilkan hasil dari no. 1, 2, dan 3.

```
[ ]  # 1. Data tuple:

     # 2. Fungsi untuk memisahkan data tuple:

     # 3. Fungsi untuk membuat dictionary:

     # 4. Hasil:
```

```
1.  ('Pensil', 1500, 'Buku', 5000, 'Penggaris', 2000)
2.  ['Pensil', 'Buku', 'Penggaris'] [1500, 5000, 2000]
3.  {'Pensil': 1500, 'Buku': 5000, 'Penggaris': 2000}
```

2. Function

2.1. Fungsi pengecekan bilangan prima (easy)

1. Tulis sebuah fungsi dalam bahasa pemrograman fungsional yang menerima sebuah angka dan mengembalikan nilai boolean True jika angka tersebut merupakan bilangan prima (hanya dapat dibagi oleh angka 1 dan bilangan itu sendiri), dan False jika tidak.

```
[ ]  # Buat fungsi disini

     is_prima(13)
```

```
True
```

2.2. Fungsi Palindrome check (hard)

2. Tulis sebuah fungsi dalam bahasa pemrograman fungsional yang menerima sebuah string dan mengembalikan nilai boolean True jika string tersebut merupakan palindrom (dapat dibaca sama dari depan dan belakang), dan False jika tidak.

```
[ ]  string = 'radar'

     # Buat fungsi disini

     isPalindrome(string)

     True
```

3. Pure Function

Ubahlah fungsi di bawah ini menjadi fungsi pure dengan cara menghilangkan penggunaan variabel global dan mengubahnya menjadi fungsi yang hanya bergantung pada inputnya

```
[ ]  total = 0

     def tambah_angka(angka):
         global total
         total += angka
         return total
```

4. Lambda

4.1. Easy

Pada soal nomor 3 terdapat pure function tambah_angka, ubahlah fungsi tersebut menjadi bentuk lambda

4.2. Hard

Ubah **salah satu** fungsi dibawah ini menjadi bentuk lambda

```
[ ]  def klasifikasi_angka(x):
         if x > 0:
             return "Positif"
         elif x < 0:
             return "Negatif"
         else:
             return "Nol"

     def pencarian_angka(list_angka,find):
         for num in list_angka:
             if num == find:
                 return "found"
         return "not found"
```

5. List Comprehension

Buat sebuah list comprehension untuk menyimpan angka ganjil antara 0 sampai 50. Dan cetak hasilnya.

```
[ ]  ganjil = # list comprehension disini
     print(ganjil)
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]
```

6. Generator Expression

Buatkan sebuah generator berdasarkan soal no 5 dan tampilkan hasilnya menggunakan iterasi.

```
[ ]  # contoh output
```

```
1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49,
```

7. HOF

Buat fungsi berikut menjadi high order function dimana fungsi sisagold akan menggunakan fungsi kurang sebagai parameternya (untuk NIM ganjil) atau sebagai return valuenya (untuk NIM genap).

```
[ ]  def sisagold(a,b):
         return a-b
```

8. Filtering

8.1. Filter list angka kelipatan 3 (easy)

0. Gunakan fungsi filter yang menerima sebuah data angka (hasil dari soal no.5 atau 6) dan mengembalikan list baru yang berisi hanya angka-angka kelipatan 3.

   contoh output :

```
[3, 9, 15, 21, 27, 33, 39, 45]
```

8.2. Filter angka prima (medium)

1. Gunakan fungsi filter yang menerima sebuah data angka (hasil dari soal no.5 atau 6) dan mengembalikan list baru yang berisi hanya angka-angka prima dari list input (gunakan fungsi prima yang kalian buat pada no.2.1 sebelumnya).

```
[ ]  data = # copas jawaban dari soal no.5 atau 6 disini

     # copas fungsi dari jawaban no.2.1 disini

     # panggil fungsi filter dan cetak hasilnya

     prima =  [1, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

8.3. Filter angka palindrome (hard)

2. Gunakan fungsi filter yang menerima sebuah kalimat yang mengandung nama lengkap kalian dan mengembalikan daftar kata yang palindrome dan yang tidak(gunakan fungsi palindrome yang kalian buat pada no.2.2 sebelumnya). Kalian bisa memanfaatkan method split string untuk memisahkan kata dalam kalimat. Serta fungsi upper atau lower untuk menyamakan huruf besar-kecil nya.

```
[ ]
    string = 'Ada apa dengan si Anna' #ganti nama anna dengan nama lengkap kalian

    # copas fungsi dari jawaban no.2.2 disini

    # panggil fungsi filter dan cetak hasilnya
```

```
data =  ['ada', 'apa', 'dengan', 'si', 'anna']
palindrome =  ['ada', 'apa', 'anna']
non-palindrome =  ['dengan', 'si']
```

9. Mapping

9.1. Opsional

Lakukan pemetaan data hasil filter dari soal no.8 sebelumnya untuk menambahkan informasi pada data sebagaimana contoh.

```
[ ]  # Contoh Output data filter soal 8.1
```

```
['3 kelipatan 3',
 '9 kelipatan 3',
 '15 kelipatan 3',
 '21 kelipatan 3',
 '27 kelipatan 3',
 '33 kelipatan 3',
 '39 kelipatan 3',
 '45 kelipatan 3']
```

```
[ ]  # Contoh Output data filter soal 8.2
```

```
['1 angka prima',
 '3 angka prima',
 '5 angka prima',
 '7 angka prima',
 '11 angka prima',
 '13 angka prima',
 '17 angka prima',
 '19 angka prima',
 '23 angka prima',
 '29 angka prima',
 '31 angka prima',
 '37 angka prima',
 '41 angka prima',
 '43 angka prima',
 '47 angka prima']
```

```
[ ]  # Contoh Output data filter soal 8.3
```

```
['ada = Palindrom', 'apa = Palindrom', 'anna = Palindrom']
```

9.2. Wajib

Susunlah baris kode berikut menjadi satu baris perintah tanpa melibatkan nama variabel apapun didalamnya.

```
[ ]  data = range(10)
     filterData = filter(lambda x : x%2==1, data)
     mapping = map(lambda y : y*2, filterData)

     [2, 6, 10, 14, 18]
```

```
[ ]  # satu baris perintah tanpa melibatkan nama variabel untuk mendapatkan output yang sama
```

10. Reduce

Gunakan reduce untuk mendapatkan hasil penjumlahan semua elemen dari data hasil filter soal no.8 sebelumnya.

```
[ ]  # import library yang dibutuhkan

     data_angka = # copy code jawaban dari soal no.8.1 disini

     # panggil fungsi reduce dan cetak hasilnya

     192
```

11. Closure

Buatlah sebuah fungsi `hitung_pangkat` yang mengimplementasikan closure di dalamnya.

12. Decorator

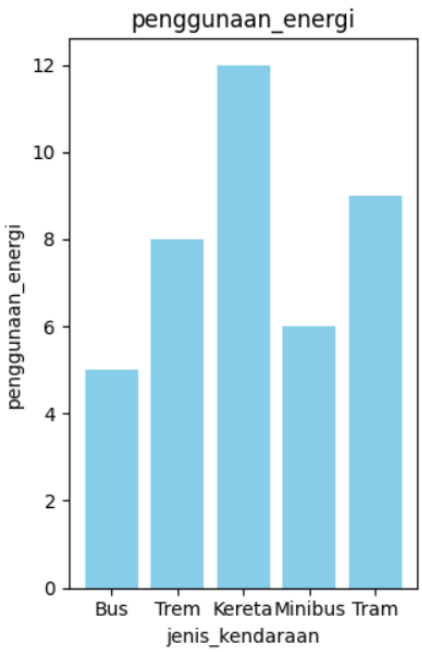
Buatlah sebuah fungsi untuk mendekorasi fungsi yang telah kalian buat pada soal no.2 sebelumnya untuk mengubah output fungsi yang sebelumnya True/False menjadi sebuah kalimat tertentu.

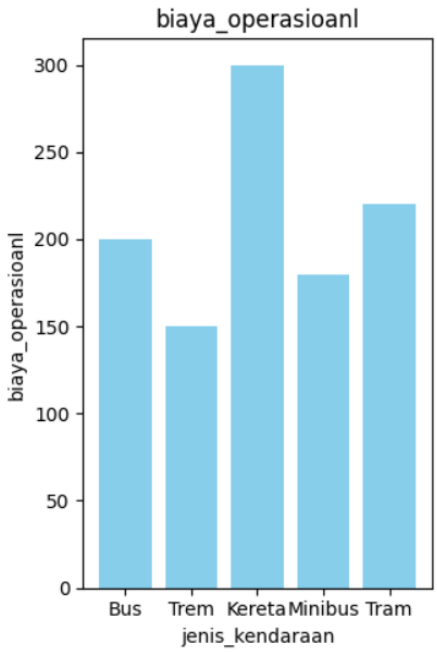
13. Data Visualization

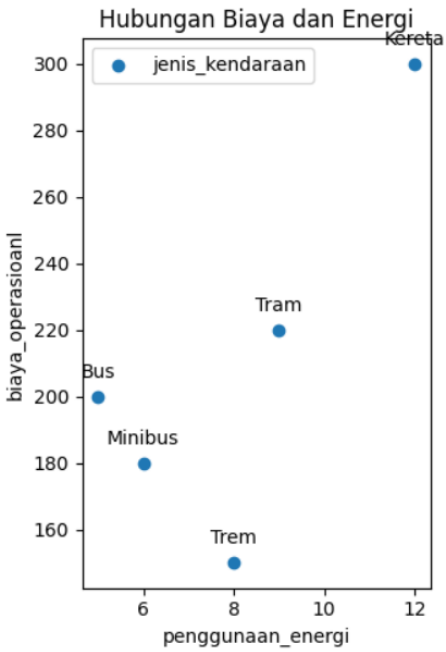
Diberikan data tentang penggunaan energi dan biaya operasional untuk berbagai jenis kendaraan umum dalam bentuk list tupel.

Buatlah visualisasi data kendaraan tentang penggunaan energi dan biaya operasional serta hubungan antara keduanya.

```
[ ]  # TODO 1: Tambahkan library yang dibutuhkan

     #Setiap tupel berisi informasi sebagai berikut:
     #(jenis_kendaraan, penggunaan_energi_per_km, biaya_operasional_per_km)
     data = [
         ('Bus', 5, 200),
         ('Trem', 8, 150),
         ('Kereta', 12, 300),
         ('Minibus', 6, 180),
         ('Tram', 9, 220)
     ]

     # TODO 2: Pisahkan Data masing-masing (dapat menggunakan pemetaan)
     jenis_kendaraan =
     penggunaan_energi =
     biaya_operasioanl =

     plt.figure(figsize=(10, 5))

     plt.subplot(1, 3, 1)
     # TODO 3: Lengkapi kode untuk visualisasi data penggunaan energi

     plt.subplot(1, 3, 2)
     # TODO 4: Lengkapi kode untuk visualisasi data biaya operasional

     plt.subplot(1, 3, 3)
     # TODO 6: Lengkapi kode untuk menggambar Scatter plot
     # TODO 7: Memberikan label pada tiap titik

     # Menambahkan legenda
     plt.legend()

     # Tampilkan plot
     plt.tight_layout()
     plt.show()
```


penggunaan_energi
penggunaan_energi
jenis_kendaraan
Bus
Trem
Kereta
Minibus
Tram
0
2
4
6
8
10
12


biaya_operasioanl
biaya_operasioanl
jenis_kendaraan
Bus
Trem
Kereta
Minibus
Tram
0
50
100
150
200
250
300


Hubungan Biaya dan Energi
jenis_kendaraan
biaya_operasioanl
penggunaan_energi
Kereta
Tram
Bus
Minibus
Trem
160
180
200
220
240
260
280
300
6
8
10
12

Rubrik Penilaian

No.	Topik	Level	Point	Easy Only	Hard-Med	Hard Only
1	Sequence		5	5	5	5
2	Function	easy	5	5		
		hard	10		10	10
3	Pure Function		5	5	5	5
4	Lambda	easy	5	5		
		hard	10		10	10
5	List Comprehension		5	5	5	5
6	Generator Expression		5	5	5	5
7	HOF		6	6	6	6
8	Filter	easy	5	5		
		medium	10		10	
		hard	15			15
9	Map	opsional	10			
		wajib	5	5	5	5
10	Reduce		7	7	7	7
11	Closure		7	7	7	7
12	Decorator		8	8	8	8
13	Data Visualization		12	12	12	12
Total			**135**	**80**	**95**	**100**