

UAS
PEMROGRAMAN DALAM SAINS

Nama : Tito Bagaswara
NIM : 20922323
Prodi : FMIPA
Jurusan : Sains Komputasi Dan Pengelolaan Energi

Soal

1. Buat suatu dictionary dengan Python yang dapat menyimpan informasi 4-5 jenis kopi dengan terlebih dahulu melakukan kuantifikasi terhadap bahan – bahan yang belum memiliki nilai, dengan satuan untuk setiap bahan adalah sama. Cantumkan satuan yang digunakan untuk jumlah bahan.
2. Konversi data harga kopi yang semula dalam USD untuk setiap pound menjadi dalam IDR untuk per kilogram dan gambarkan grafiknya untuk suatu rentang waktu tertentu.
3. Buat fungsi – fungsi untuk menghasilkan elemen – elemen matriks A dan B dan buatlah kedua matriks tersebut.
4. Gambarkan grafik prediksi harga kopi untuk tahun 2024. Jelaskan alasan mengapa digunakan rentang data tertentu, Perata – rataan pada kurun tertentu dan lainnya sehingga dapat memperkuat pendekatan yang diambil.
5. Dengan memanfaatkan dictionary dan prediksi harga kopi, tentukan jumlah 4-5 produk yang dipilih sehingga memberikan keuntungan optimum suatu kedai kopi di suatu waktu pada tahun 2024.

Jawab

1. Dictionary dengan Python yang dapat menyimpan informasi 4-5 jenis kopi

Ingredients	Short Black Espresso	Macchiato	Americano	Cappuccino	Mocha
Coffee	30 ml	30 ml	30 ml	30 ml	30 ml
Milk	0 ml	10 ml (steamed/foam)	0 ml	150 ml (Steam/Foam)	150 ml (steamed)
Sugar	0 g	25 g (as necessary)	0 g	25 g	10 g
Chocolate	0 g	0 g	0 g	1 g (Sprinkle)	10 g (melted)
Water	0 ml	0 ml	90 ml (hot)	0 ml	0 ml
Cream	0 ml	0 ml	0 ml	0 ml	15 ml
Ice cream	0 g	0 g	0 g	0 g	0 g
Ice	0	0	0	0	0

Python code :

```
from tabulate import tabulate

# Membuat dictionary untuk menyimpan informasi jenis kopi
jenis_kopi = {
    'Short Black Espresso': {
```

```
'Coffee': '30 ml',
'Milk': '0 ml',
'Sugar': '0 g',
'Chocolate': '0 g',
'Water': '0 ml',
'Cream': '0 ml',
'Ice cream': '0 g',
'Ice': '0'
},
'Macchiato': {
  'Coffee': '30 ml',
  'Milk': '10 ml (steamed/foam)',
  'Sugar': '25 g (as necessary)',
  'Chocolate': '0 g',
  'Water': '0 ml',
  'Cream': '0 ml',
  'Ice cream': '0 g',
  'Ice': '0'
},
'Americano': {
  'Coffee': '30 ml',
  'Milk': '0 ml',
  'Sugar': '0 g',
  'Chocolate': '0 g',
  'Water': '90 ml (hot)',
  'Cream': '0 ml',
  'Ice cream': '0 g',
  'Ice': '0'
},
'Cappuccino': {
  'Coffee': '30 ml',
  'Milk': '150 ml (Steam/Foam)',
  'Sugar': '25 g',
  'Chocolate': '1 g (Sprinkle)',
  'Water': '0 ml',
  'Cream': '0 ml',
  'Ice cream': '0 g',
  'Ice': '0'
},
'Mocha': {
  'Coffee': '30 ml',
  'Milk': '150 ml (steamed)',
  'Sugar': '10 g',
  'Chocolate': '10 g (melted)',
  'Water': '0 ml',
  'Cream': '15 ml',
  'Ice cream': '0 g',
  'Ice': '0'
```

```

    }
}

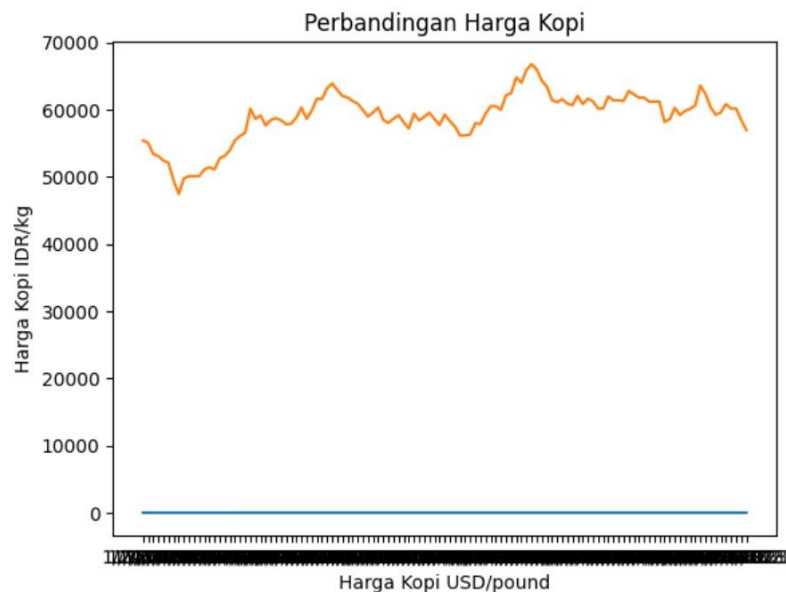
# Mengkonversi dictionary menjadi list of lists
table_data = []
header = ['Ingredients'] + list(jenis_kopi.keys())
for ingredient in jenis_kopi['Short Black Espresso']:
    row = [ingredient]
    for jenis in jenis_kopi:
        row.append(jenis_kopi[jenis].get(ingredient, '-'))
    table_data.append(row)

# Menghasilkan tabel dengan menggunakan tabulate
table = tabulate(table_data, headers=header, tablefmt='grid')

# Menampilkan tabel
print(table)

```

2. Tabel harga kopi dari USD/Pound ke IDR/Kg dengan asumsi menggunakan harga per tahun 2023



Python :

```

import pandas as pd
import matplotlib.pyplot as plt

# Membaca file CSV
data = pd.read_csv('Perbandingan harga kopi.csv')

# Mengambil data kolom tertentu

```

```

kolom1 = data['Month']
kolom2 = data['Price USD/pound']
kolom3 = data['Price IDR/kg']

```

```

# Menampilkan grafik
plt.plot(kolom1, kolom2)
plt.xlabel('Harga Kopi USD/pound')
plt.ylabel('Harga Kopi IDR/kg')
plt.title('Perbandingan Harga Kopi')
plt.legend()
plt.show()

```

Month	Price USD/pound	Price IDR/kg
02/01/2023	1,67	55.406,58
03/01/2023	1,66	55.074,80
04/01/2023	1,61	53.415,92
05/01/2023	1,6	53.084,15
06/01/2023	1,58	52.420,60
09/01/2023	1,57	52.088,82
10/01/2023	1,49	49.434,61
11/01/2023	1,43	47.443,96
12/01/2023	1,5	49.766,39
13/01/2023	1,51	50.098,16
16/01/2023	1,51	50.098,16
17/01/2023	1,51	50.098,16
18/01/2023	1,54	51.093,49
19/01/2023	1,55	51.425,27
20/01/2023	1,54	51.093,49
23/01/2023	1,59	52.752,37
24/01/2023	1,6015	53.133,91
25/01/2023	1,627	53.979,94
26/01/2023	1,6709	55.436,44
27/01/2023	1,6908	56.096,67
30/01/2023	1,7053	56.577,75
31/01/2023	1,8129	60.147,66
01/02/2023	1,7671	58.628,12
02/02/2023	1,7827	59.145,69
03/02/2023	1,7378	57.656,02
06/02/2023	1,7621	58.462,23
07/02/2023	1,7705	58.740,93
08/02/2023	1,7608	58.419,10
09/02/2023	1,7428	57.821,91
10/02/2023	1,7461	57.931,39

13/02/2023	1,7727	58.813,92
14/02/2023	1,8179	60.313,54
15/02/2023	1,7678	58.651,35
16/02/2023	1,8038	59.845,74
17/02/2023	1,8567	61.600,84
20/02/2023	1,8567	61.600,84
21/02/2023	1,9039	63.166,82
22/02/2023	1,9264	63.913,31
23/02/2023	1,8966	62.924,62
24/02/2023	1,8698	62.035,46
27/02/2023	1,8635	61.826,44
28/02/2023	1,8477	61.302,24
01/03/2023	1,8353	60.890,83
02/03/2023	1,807	59.951,91
03/03/2023	1,7782	58.996,39
06/03/2023	1,7959	59.583,64
07/03/2023	1,8183	60.326,82
08/03/2023	1,7623	58.468,87
09/03/2023	1,7496	58.047,51
10/03/2023	1,768	58.657,98
13/03/2023	1,7839	59.185,51
14/03/2023	1,7511	58.097,28
15/03/2023	1,7244	57.211,44
16/03/2023	1,7899	59.384,57
17/03/2023	1,7601	58.395,88
20/03/2023	1,7766	58.943,31
21/03/2023	1,7943	59.530,55
22/03/2023	1,7669	58.621,49
23/03/2023	1,7399	57.725,69
24/03/2023	1,7861	59.258,50
27/03/2023	1,7579	58.322,89
28/03/2023	1,7336	57.516,67
29/03/2023	1,6926	56.156,39
30/03/2023	1,6931	56.172,98
31/03/2023	1,6971	56.305,69
03/04/2023	1,7475	57.977,84
04/04/2023	1,7442	57.868,36
05/04/2023	1,7903	59.397,84
06/04/2023	1,8241	60.519,25
07/04/2023	1,8241	60.519,25
10/04/2023	1,8076	59.971,82
11/04/2023	1,8736	62.161,54
12/04/2023	1,8818	62.433,59
13/04/2023	1,9535	64.812,43
14/04/2023	1,929	63.999,57
17/04/2023	1,9868	65.917,24

18/04/2023	2,0127	66.776,54
19/04/2023	1,989	65.990,23
20/04/2023	1,9392	64.337,99
21/04/2023	1,9114	63.415,65
28/04/2023	1,8512	61.418,36
01/05/2023	1,8417	61.103,17
02/05/2023	1,856	61.577,61
03/05/2023	1,8358	60.907,42
04/05/2023	1,8284	60.661,91
05/05/2023	1,8703	62.052,05
08/05/2023	1,8337	60.837,75
09/05/2023	1,8578	61.637,33
10/05/2023	1,8457	61.235,88
11/05/2023	1,8137	60.174,20
12/05/2023	1,8141	60.187,47
15/05/2023	1,8681	61.979,06
16/05/2023	1,8505	61.395,13
17/05/2023	1,8508	61.405,09
18/05/2023	1,8475	61.295,60
19/05/2023	1,8925	62.788,59
22/05/2023	1,877	62.274,34
23/05/2023	1,8627	61.799,90
24/05/2023	1,862	61.776,68
25/05/2023	1,8445	61.196,07
26/05/2023	1,8445	61.196,07
29/05/2023	1,8445	61.196,07
30/05/2023	1,7539	58.190,18
31/05/2023	1,7648	58.551,81
01/06/2023	1,8169	60.280,37
02/06/2023	1,7841	59.192,14
05/06/2023	1,803	59.819,20
06/06/2023	1,812	60.117,80
07/06/2023	1,8271	60.618,78
08/06/2023	1,917	63.601,44
09/06/2023	1,8794	62.353,97
12/06/2023	1,8199	60.379,90
13/06/2023	1,7857	59.245,23
14/06/2023	1,7959	59.583,64
15/06/2023	1,833	60.814,53
16/06/2023	1,8134	60.164,25
19/06/2023	1,8134	60.164,25
20/06/2023	1,7619	58.455,60
21/06/2023	1,7174	56.979,20

3. Fungsi untuk menghasilkan elemen – elemen matriks A dan B

```

#fungsi ini adalah untuk menghitung elemen matriks a
def elemen_a(p,q,x,y):
    sum=0
    for xi in x:
        sum+= xi**(p+q-2)

    return sum

#fungsi ini adalah untuk menghitung elemen matriks b
def elemen_b(p,q,x,y):
    sum=0
    n=len(x)
    for i in range(n):
        sum+=y[i]*x[i]**(p+q-2)

    return sum

#fungsi ini adalah untuk membuat matriks A
def create_matriks_A(m,x,y):
    mat=[]

    for p in range(m+1):
        row=[]
        for q in range(m+1):
            apq=elemen_a(p+1, q+1, x, y)
            row.append(apq)

        mat.append(row)

    return mat

#fungsi ini adalah untuk mencetak matriks A per baris
def print_matriks_A(mat):
    for row in mat:
        print(row)

#fungsi ini adalah untuk membuat matriks B
def create_matriks_B(m,x,y):
    mat=[]

    for p in range(m+1):
        apq=elemen_b(p+1, 1, x, y)

```

```
        mat.append(apq)

    return mat

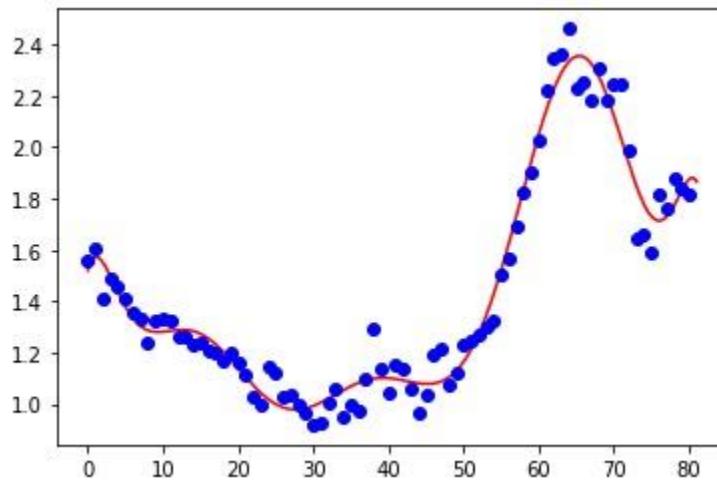
#fungsi ini adalah untuk mencetak matriks B per baris
def print_matriks_B(mat):
    for bi in mat:
        print(bi)

#Fungsi untuk mencari matriks C atau matriks regresi
def ymodel(xx,C):

    yy=[]
    for x in xx:
        sum=0;
        for j in range(len(C)):
            sum+=C[j]*x**j
        yy.append(sum)

    return yy
```


4. Gambarkan grafik prediksi harga kopi untuk tahun 2024. Jelaskan alasan mengapa digunakan rentang data tertentu, Perata - rataan pada kurun tertentu dan lainnya sehingga dapat memperkuat pendekatan yang diambil.



$$f(x) = 1.52024600 + 1.03440105e^{-1}x - 6.22958370e^{-2}x^2 + 1.08935610e^{-2}x^3 - 9.39777974e^{-4}x^4 + 4.59464843e^{-5}x^5 - 1.35427567e^{-6}x^6 + 2.45047945e^{-8}x^7 - 2.65971903e^{-10}x^8 + 1.58736673e^{-12}x^9 - 4.00136811e^{-15}x^{10}$$

HARGA PREDIKSI PER POUND (USD)	0,956836
HARGA PREDIKSI PER KG (RP)	31.453