

# LAPORAN UAS BAGIAN-1



**SK5003-Pemrograman Dalam Sains**

**Program Studi Magister Sains Komputasi  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Bandung**

**Anggota Kelompok :**

Alief Pascal Taruna-20922321

Aziz Mustika Aji-20922316

Galih Arisona-20922325

Toro Rahman Aziz-20922322

## 1. Latar Belakang

Dalam rangka penyelesaian tugas UAS yang melibatkan pembuatan *dictionary* untuk menyimpan informasi kopi, konversi harga kopi, dan pembuatan fungsi untuk matriks, latar belakang dapat dirumuskan sebagai berikut:

Pada ujian ini, fokusnya adalah pada pemrosesan data kopi dan pengolahan informasi terkait. Kopi adalah salah satu komoditas yang memiliki peran penting dalam industri dan pasar global. Informasi yang terkait dengan kopi, seperti bahan-bahan yang digunakan, konversi harga, dan manipulasi matriks, dapat membantu dalam pengambilan keputusan yang lebih baik dalam hal produksi, distribusi, dan analisis komoditas kopi.

Untuk memulai tugas ini, kita akan membuat sebuah *dictionary* dengan menggunakan bahasa pemrograman Python. *Dictionary* ini akan digunakan untuk menyimpan informasi tentang kopi, seperti jenis kopi, jumlah bahan yang digunakan, dan satuan yang digunakan untuk setiap bahan.

Selanjutnya, kita akan melibatkan konversi data harga kopi dari (USD) per *pound* menjadi rupiah (IDR) per kilogram (kg).

Terakhir, kita akan membuat beberapa fungsi untuk menghasilkan elemen-elemen matriks A dan B. Matriks A dan B dapat digunakan dalam berbagai aplikasi, seperti pemodelan matematika, analisis data, atau algoritma pengolahan citra. Dalam tugas ini, kita akan fokus pada pembuatan prediksi harga kopi untuk tahun 2024.

## 2. Tujuan

Menyelesaikan tugas UAS bagian satu sebagai berikut:

**2.1** Membuat suatu *dictionary* dengan python yang dapat menyimpan 4-5 jenis informasi 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.

**2.2** Konversi data harga kopi yang semula dalam USD untuk setiap pound menjadi IDR per kilogram dan gambarkan grafiknya untuk suatu rentang waktu tertentu.

**2.3** Membuat fungsi untuk menghasilkan elemen-elemen matriks A dan B buatlah kedua matriks tersebut.

### 3. Hasil Jawaban

Pada bab hasil analisis ini memaparkan hasil jawaban pada bagian bab 2 tujuan. Pertama, kami membuat *dictionary* menggunakan Python untuk menyimpan informasi kopi. Kedua, kami melakukan konversi data harga kopi dari USD per pound menjadi IDR per kg dan membuat grafik visualisasi perubahan harga kopi dalam rentang waktu tertentu. Terakhir, kami berhasil membuat fungsi untuk menghasilkan elemen-elemen matriks A dan B.

#### 3.1 Dictionary Informasi Kopi

```
# Dictionary of 5 types of coffee  
# Unit for Ingredient = milliliters
```

```
import pandas as pd  
  
coffee_types = {  
    'irish': {  
        'coffee': 30,  
        'milk': 0,  
        'sugar': 30,  
        'chocolate': 0,  
        'water': 10,  
        'cream': 45,  
        'ice_cream': 0,  
        'whiskey': 150,  
        'ice': 0  
    },  
    'doppio': {  
        'coffee': 60,  
        'milk': 0,  
        'sugar': 0,  
        'chocolate': 0,  
        'water': 0,  
        'cream': 0,  
        'ice_cream': 0,  
        'whiskey': 0,  
        'ice': 0  
    },  
    'cappuccino': {  
        'coffee': 30,  
        'milk': 150,  
        'sugar': 0,  
        'chocolate': 2,  
        'water': 0,  
        'cream': 0,  
        'ice_cream': 0,  
    }  
}
```

```
        'whiskey': 0,  
        'ice': 0  
    },  
    'breve': {  
        'coffee': 60,  
        'milk': 45,  
        'sugar': 0,  
        'chocolate': 0,  
        'water': 0,  
        'cream': 45,  
        'ice_cream': 0,  
        'whiskey': 0,  
        'ice': 0  
    },  
    'mocha': {  
        'coffee': 30,  
        'milk': 150,  
        'sugar': 0,  
        'chocolate': 16,  
        'water': 0,  
        'cream': 0,  
        'ice_cream': 0,  
        'whiskey': 0,  
        'ice': 0  
    }  
}  
  
tableDictionary = pd.DataFrame.from_dict(coffee_types, orient='index')  
  
print(tableDictionary)
```

	☪ coffee	☪ milk	☪ sugar	☪ chocolate	☪ water	☪ cream	☪ ice_cream	☪ whiskey	☪ ice
irish	30	0	30	0	10	45	0	150	0
doppio	60	0	0	0	0	0	0	0	0
cappuccino	30	150	0	2	0	0	0	0	0
breve	60	45	0	0	0	45	0	0	0
mocha	30	150	0	16	0	0	0	0	0

### 3.2 Konversi Data Kopi dan Grafik

# Konversi dari USD/pound menjadi IDR/kg

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('coffee-price-2023.csv')

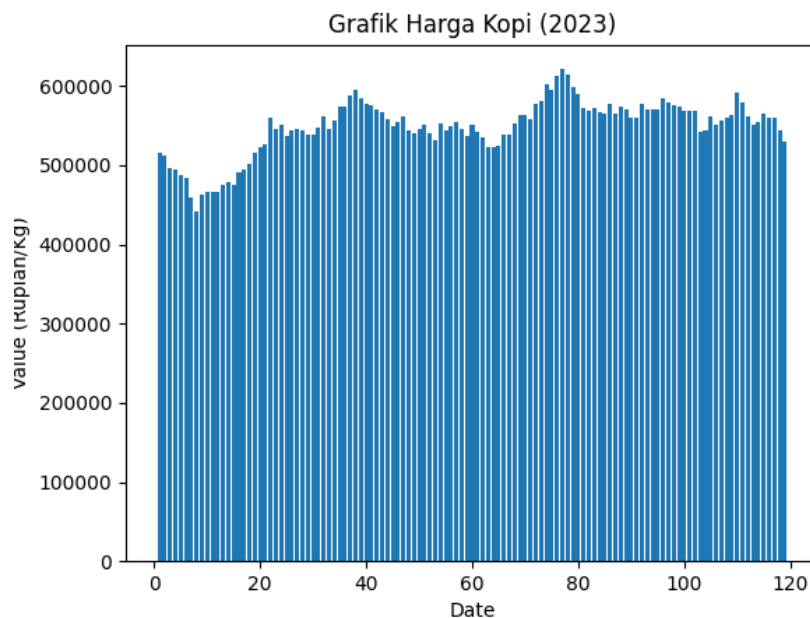
harga = data['value'].to_list()
kurs_konversi = 14000 # Misalnya, kurs konversi 1 dollar = 14.000 Rupiah
massa_konversi = 0.453592 # Konversi dari Pound ke KG

harga_rupiah = [harga * (kurs_konversi / massa_konversi) for harga in harga]
rounded_harga_rupiah = [round(harga, 1) for harga in harga_rupiah]

# Membuat label sumbu x dari 1 sampai dengan panjang data harga
x_labels = range(1, len(rounded_harga_rupiah) + 1)

# Membuat bar chart
plt.bar(x_labels, rounded_harga_rupiah)
plt.xlabel('Date')
plt.ylabel('Value (Rupiah/Kg)')
plt.title('Grafik Harga Kopi (2023)')

# Menampilkan grafik
plt.show()
```



### 3.3 Fungsi Untuk Hasilkan Elemen Matriks A dan B

# Method untuk membuat elemen dari matriks A dan B

# Membuat elemen untuk matriks A

```
def elemA(p, q, x, y):  
    sigma = 0  
    for xi in x:  
        sigma += xi ** (p + q - 2)  
    return sigma
```

# Membuat elemen untuk matriks B

```
def elemB(p, q, x, y):  
    sigma = 0  
    N = len(x)  
    for i in range(N):  
        sigma += y[i] * x[i] ** (p + q - 2)  
    return sigma
```

# Membuat matriks A

```
def createMatrixA(M, x, y):  
    mat = []  
    for p in range(M + 1):  
        row = []  
        for q in range(M + 1):  
            # Python list begin from 0, so we have to add 1 for each p and q  
            apq = elemA(p + 1, q + 1, x, y)  
            row.append(apq)  
        mat.append(row)  
    return mat
```

# Membuat matriks B

```
def createMatrixB(M, x, y):  
    mat = []  
    for p in range(M + 1):  
        row = []  
        # Python list begin from 0, so we have to add 1 for each p and q  
        bpq = elemB(p + 1, 1, x, y)  
        row.append(bpq)  
        mat.append(row)  
    return mat
```