

PRAKTIKUM PEMROGRAMAN KOMPUTER

PERTEMUAN IX

ARRAY MULTIDIMENSI



Disusun oleh:

Nama : Ahmad Fawwas Dzulfahmi
NIM : 25/567941/SV/27316
Kelas : RI1A2
Dosen Pengampu : Ir. Yuris Mulya Saputra, S.T., M.Sc., Ph.D., IPM.

PROGRAM STUDI D-IV TEKNOLOGI REKAYASA INTERNET

DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS GADJAH MADA

2025

DAFTAR ISI

DAFTAR ISI.....	1
DAFTAR GAMBAR.....	2
BAB I PENDAHULUAN.....	3
1.1 Latar Belakang.....	3
1.3 Tujuan.....	3
BAB II PEMBAHASAN.....	4
2.1 Tugas 1.....	4
2.2 Tugas 2.....	6
2.3 Tugas 3.....	7
2.4 Tugas 4.....	9
BAB III PENUTUP.....	11
3.1 Kesimpulan.....	11
DAFTAR PUSTAKA.....	12

DAFTAR GAMBAR

Tugas 1 : Source Code list dan array	4
Tugas 1.1 : perintah menampilkan elemen pertama	4
Tugas 1.1 : output perintah menampilkan elemen pertama	4
Tugas 1.2 : perintah menampilkan elemen kolom terakhir	5
Tugas 1.2 : output menampilkan elemen kolom terakhir	5
Tugas 2.1 : source code nested list comprehension	6
Tugas 2.2 : Output kode program nested list comprehension	6
Tugas 3 : source code soal matrik.....	7
Tugas 3.1 : Source Code program penjumlahan dan pengurangan numpy	7
Tugas 3.1 : output program penjumlahan dan pengurangan numpy.....	7
Tugas 3.2 : source code perkalian numpy	8
Tugas 3.2 : output perkalian numpy	8
Tugas 4 : data array 2D nilai mahasiswa	9
Tugas 4.1 : source code menampilkan nilai mahasiswa	9
Tugas 4.1 : output program menampilkan nilai.....	9
Tugas 4.2 : source code program menghitung nilai rata-rata, tertinggi dan terendah.....	10
Tugas 4.2 : output program menghitung nilai rata-rata, tertinggi dan terendah	10

BAB I

PENDAHULUAN

1.1 Latar Belakang

Array multidimensi merupakan pengembangan dari konsep array satu dimensi yang sebelumnya telah dipelajari. Pada array multidimensi, setiap elemen dapat menyimpan lebih dari satu nilai dengan struktur baris dan kolom. Hal ini memungkinkan pengolahan data yang lebih kompleks dan teratur, seperti data tabel, nilai mahasiswa, atau bentuk matriks yang tersusun secara sistematis.

Array multidimensi secara umum direpresentasikan menggunakan struktur list bersarang (nested list). Struktur ini memungkinkan sebuah list untuk berisi list-list lain, di mana setiap tingkat kedalaman list tersebut merepresentasikan satu dimensi tambahan. Pendekatan list bersarang ini juga merupakan cara standar untuk membuat matriks, yaitu kumpulan elemen dalam bentuk baris dan kolom, karena Python sendiri tidak memiliki tipe data bawaan khusus untuk matriks.

Dalam Python, array multidimensi dapat dibuat menggunakan list bersarang, perulangan (looping), maupun library NumPy. NumPy menjadi sangat penting karena library ini memudahkan berbagai operasi aljabar linear. Operasi matematis pada matriks seperti penjumlahan, pengurangan, perkalian, dan transpose dapat dilakukan secara lebih efisien dan sederhana dibandingkan dengan implementasi manual yang kompleks menggunakan list bersarang.

1.2 Rumusan Masalah

Adapun Rumusan Masalah Sebagai Berikut:

1. Membuat sebuah program untuk menampilkan elemen tertentu dari array 3D.
2. Membuat sebuah program nested list comprehension untuk membuat matriks.
3. Membuat sebuah program untuk melakukan operasi aljabar sederhana.
4. Membuat sebuah program untuk menampilkan dan mengolah nilai mahasiswa.

1.3 Tujuan

Adapun Tujuan dari Laporan Praktikum sebagai berikut:

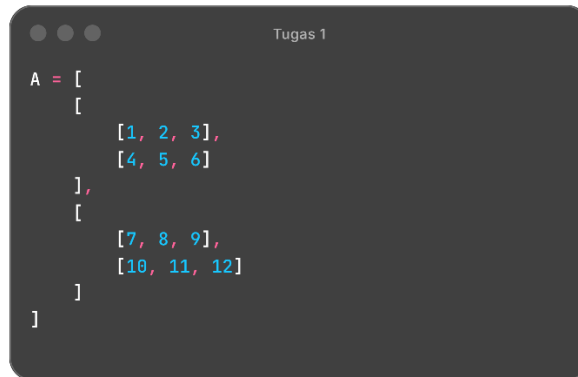
1. Mahasiswa mampu menjelaskan konsep array dua dimensi dan multidimensi dalam Python.
2. Mahasiswa mampu menjelaskan cara deklarasi dan pengaksesan elemen array multidimensi.
3. Mahasiswa mampu mengimplementasikan operasi dasar pada array multidimensi seperti penjumlahan, pengurangan, dan transpose.
4. Mahasiswa mampu menggunakan library NumPy dalam array multidimensi.

BAB II

PEMBAHASAN

2.1 Tugas 1

1. Diberikan array 3 dimensi berikut:



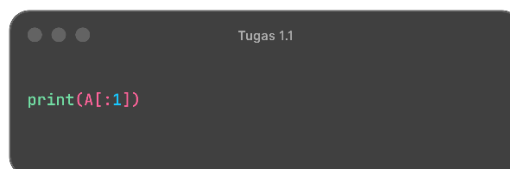
```
A = [  
  [  
    [1, 2, 3],  
    [4, 5, 6]  
  ],  
  [  
    [7, 8, 9],  
    [10, 11, 12]  
  ]  
]
```

Tugas 1 : Source Code list dan array

Lakukan perintah berikut dan jelaskan :

- Tampilkan semua elemen pada lapisan pertama saja (menggunakan slicing).
- Tampilkan semua elemen kolom terakhir dari setiap baris dan lapisan

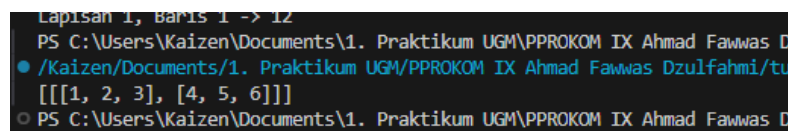
Jawaban :



```
print(A[:1])
```

Tugas 1.1 : perintah menampilkan elemen pertama

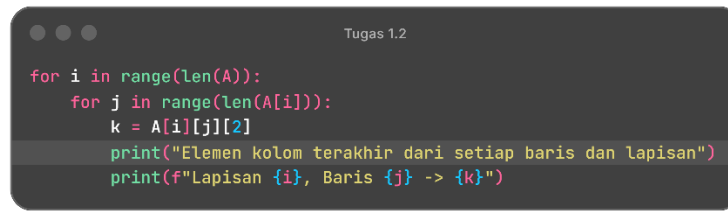
Untuk perintah a cukup mudah karena kita tinggal menuliskan perintah di atas, atau juga bisa mengganti `[:1]` dengan `[0]`. Tapi karena perintahnya menggunakan slicing kita menggunakan `[:1]`, artinya semua indeks dari 0 samapi 1 akan ditampilkan. Untuk outputnya akan sama seperti pada gambar *tugas 1.1 : output perintah menampilkan elemen pertama*



```
Lapisan 1, Baris 1 -> 12  
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas D  
● /Kaizen/Documents/1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi/tu  
[[[1, 2, 3], [4, 5, 6]]]  
○ PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas D
```

Tugas 1.1 : output perintah menampilkan elemen pertama

juga kalau append di array kita hanya bisa memasukan sesuai dengan simbol yang tertera, contoh seperti pada baris ke tiga. Sedangkan untuk list kita bisa memasukan data secara bebas.

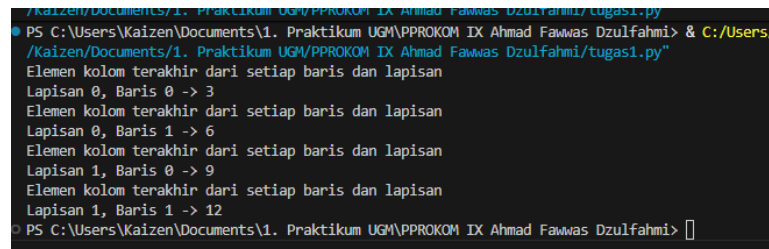


```
Tugas 1.2

for i in range(len(A)):
    for j in range(len(A[i])):
        k = A[i][j][2]
        print("Elemen kolom terakhir dari setiap baris dan lapisan")
        print(f"Lapisan {i}, Baris {j} -> {k}")
```

Tugas 1.2 : perintah menampilkan elemen kolom terakhir

Untuk perintah kedua kita diminta untuk menampilkan semua elemen di kolom terakhir. Struktur perulangan pertama dengan `for i in range(len(A))` berfungsi untuk mengakses setiap lapisan dalam array, sedangkan perulangan kedua dengan `for j in range(len(A[i]))` digunakan untuk mengakses setiap baris pada lapisan ke-i. Di dalamnya terdapat perintah `k = A[i][j][2]` yang mengambil elemen pada indeks kolom ke-2 atau elemen terakhir dari setiap baris, karena setiap baris berisi tiga nilai. Nilai yang diambil kemudian ditampilkan menggunakan perintah `print(f"Lapisan {i}, Baris {j} -> {k}")`, sehingga menampilkan posisi lapisan, baris, serta elemen terakhir dari masing-masing bagian. Dan untuk outputnya akan seperti dibawah



```
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi> python /Kaizen/Documents/1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi/tugas1.py
Elemen kolom terakhir dari setiap baris dan lapisan
Lapisan 0, Baris 0 -> 3
Elemen kolom terakhir dari setiap baris dan lapisan
Lapisan 0, Baris 1 -> 6
Elemen kolom terakhir dari setiap baris dan lapisan
Lapisan 1, Baris 0 -> 9
Elemen kolom terakhir dari setiap baris dan lapisan
Lapisan 1, Baris 1 -> 12
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi>
```

Tugas 1.2 : output menampilkan elemen kolom terakhir

2.2 Tugas 2

2. Berikut adalah matriks identitas dengan ukuran 4 x 4 :

```
[
  [1, 0, 0, 0],
  [0, 1, 0, 0],
  [0, 0, 1, 0],
  [0, 0, 0, 1]
]
```

Lakukan perintah berikut dan jelaskan :

- Gunakan nested list comprehension untuk membuat pola matriks identitas di atas
- Tambahkan variabel input n, agar pengguna bisa menentukan ukuran matriks identitas dan menampilkan matriks yang baru

Jawaban :

```
Tugas 2.1

n = int(input("Masukkan panjang kolom dan baris : "))
identity = [[0] * i + [1] + [0] * (n - i - 1) for i in range(n)]

for i in identity:
    print(i)
```

Tugas 2.1 : source code nested list comprehension

```
Lapisan 1, Baris 1 -> 12
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi> & C:\
/Kaizen/Documents/1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi\tugas2.py"
Masukkan panjang kolom dan baris : 4
[1, 0, 0, 0]
[0, 1, 0, 0]
[0, 0, 1, 0]
[0, 0, 0, 1]
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi> []
```

Tugas 2.2 : Output kode program nested list comprehension

Penjelasan :

Nested list comprehension adalah perintah untuk membuat list di dalam list dengan perintah ringkas didalam satu baris kode kalau memungkinkan. Jadi pada baris pertama meminta input berupa panjang kolom dan baris melalui variabel n. Kemudian pada baris kedua, variabel identity dibuat menggunakan list comprehension dengan pola `[[0] * i + [1] + [0] * (n - i - 1) for i in range(n)]`. Ekspresi tersebut berarti setiap baris matriks dibentuk dari sejumlah nol di sebelah kiri (`[0] * i`), satu angka satu di tengah (`[1]`), dan sisanya nol di sebelah kanan (`[0] * (n - i - 1)`). Proses ini diulang sebanyak n kali sesuai jumlah baris yang dimasukkan pengguna, sehingga membentuk pola diagonal dari angka 1 yang menjadi ciri khas matriks identitas. Terakhir, perulangan `for i in identity:` digunakan untuk menampilkan setiap baris matriks identitas secara berurutan ke layar.

2.3 Tugas 3

3. Diketahui dua matriks berikut:

```
Tugas 3

A = [
  [2, 4, 6],
  [1, 3, 5]
]
B = [
  [1, 1, 1],
  [2, 2, 2]
]
```

Tugas 3 : source code soal matrik

Lakukan perintah berikut dan jelaskan :

- Lakukan penjumlahan dan pengurangan kedua matriks di atas dengan NumPy.
- Hitung perkalian matriks $A \times B^T$.

Jawaban :

```
Tugas 3.1

import numpy as np
A = np.array([
  [2, 4, 6],
  [1, 3, 5]
])

B = np.array([
  [1, 1, 1],
  [2, 2, 2]
])

penjumlahan = A + B
pengurangan = A - B

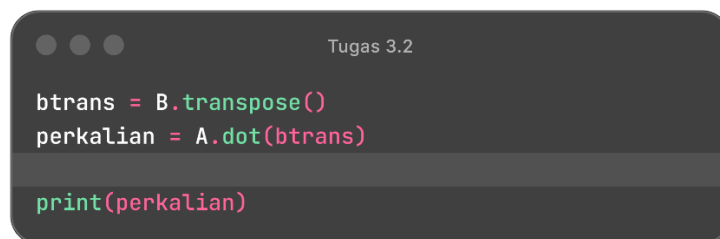
print(penjumlahan)
print(pengurangan)
```

Tugas 3.1 : Source Code program penjumlahan dan pengurangan numpy

```
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi\tugas3>
[[3 5 7]
 [3 5 7]]
[[ 1  3  5]
 [-1  1  3]]
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi\tugas3>
```

Tugas 3.1 : output program penjumlahan dan pengurangan numpy

Untuk melakukan penjumlahan dan perhitungan kita pertama import terlebih dahulu library numpy. Lalu pada bagian awal, dua array dua dimensi didefinisikan, yaitu A dan B, masing-masing dibuat menggunakan fungsi np.array(). Matriks A berisi nilai $\begin{bmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 \end{bmatrix}$, sedangkan matriks B berisi nilai $\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix}$. Dengan menggunakan operasi langsung $A + B$, Python akan melakukan penjumlahan elemen-elemen pada posisi yang sama dari kedua matriks tersebut, sehingga menghasilkan matriks baru hasil penjumlahan. Begitu pula pada operasi $A - B$, setiap elemen pada matriks B dikurangkan dari elemen yang sesuai pada matriks A, menghasilkan matriks baru hasil pengurangan. Proses ini dapat dilakukan dengan sangat sederhana karena NumPy secara otomatis mengenali bentuk dan ukuran kedua matriks sehingga mampu melakukan operasi aritmatika elemen-per-elemen tanpa perlu menggunakan perulangan manual. Outputnya akan terlihat seperti pada gambar *Tugas 3.2 : output program penjumlahan dan pengurangan numpy*

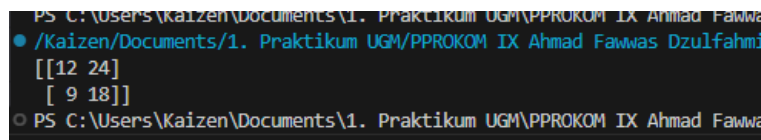


```
Tugas 3.2

btrans = B.transpose()
perkalian = A.dot(btrans)

print(perkalian)
```

Tugas 3.2 : source code perkalian numpy



```
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfa...
● /Kaizen/Documents/1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfa...
[[12 24]
 [ 9 18]]
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfa...
```

Tugas 3.2 : output perkalian numpy

Untuk melakukan perkalian kita bisa melakukannya dengan numpy juga, cukup tambahkan baris kode seperti di atas. Pada baris pertama, perintah `btrans = B.transpose()` digunakan untuk melakukan transpose terhadap matriks B, yaitu menukar posisi baris menjadi kolom dan kolom menjadi baris. Hasil transpose ini kemudian disimpan ke dalam variabel `btrans`. Setelah itu, dilakukan operasi perkalian matriks dengan perintah `A.dot(btrans)`. Fungsi `dot()` digunakan untuk menghitung hasil kali antara dua matriks sesuai aturan aljabar linear, di mana jumlah kolom pada matriks pertama harus sama dengan jumlah baris pada matriks kedua. Hasil dari operasi ini disimpan pada variabel `perkalian` dan kemudian ditampilkan dengan perintah `print(perkalian)`. Dengan demikian, program ini menghasilkan matriks baru yang merupakan hasil perkalian antara matriks A dan matriks B yang telah ditranspose. Dan hasilnya seperti pada gambar *tugas 3.2 : output perkalian numpy*

2.4 Tugas 4

4. Seorang dosen menyimpan data nilai 24 mahasiswa kelas A1 dalam 3 kolom, yakni Tugas, UTS, dan UAS dengan indeks berurutan. Nilai disimpan dalam array 2D seperti berikut:

```
Tugas 4

nilai = [
    [85, 80, 90],
    [78, 82, 88],
    [92, 90, 94],
    [70, 68, 72],
    [88, 85, 84],
    [60, 75, 70],
    [95, 92, 98],
    [74, 70, 76],
    [81, 85, 83],
    [69, 72, 70],
    [90, 88, 92],
    [76, 80, 79],
    [84, 86, 90],
    [79, 82, 85],
    [67, 70, 68],
    [91, 94, 93],
    [73, 78, 75],
    [87, 84, 89],
    [65, 68, 70],
    [93, 90, 95],
    [77, 80, 78],
    [82, 84, 88],
    [89, 85, 90],
    [71, 74, 76]
]
```

Tugas 4 : data array 2D nilai mahasiswa

Jawaban :

```
Tugas 4.1

import numpy as np

nilai = [
    [85, 80, 90],
    [78, 82, 88],
    [92, 90, 94],
    [70, 68, 72],
    .....
]

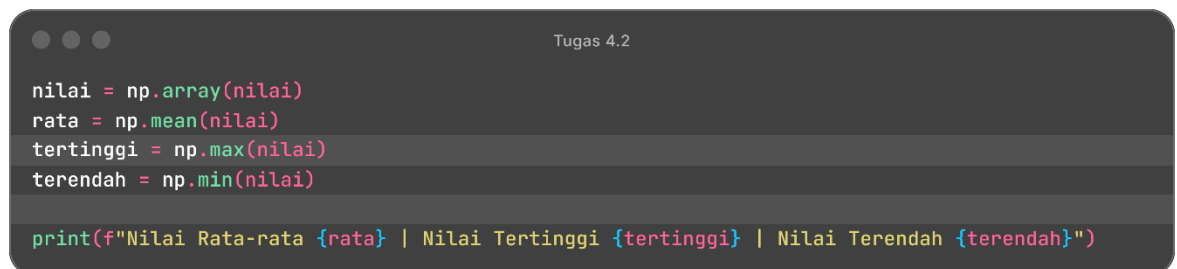
for i in range(len(nilai)):
    print(f"Mahasiswa ke-{i+1} | Tugas: {nilai[i][0]} | UTS: {nilai[i][1]} | UAS: {nilai[i][2]}")
```

Tugas 4.1 : source code menampilkan nilai mahasiswa

```
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi> & C:/Users/Kaizen/Documents/1. Praktikum UGM/PPROKOM IX Ahmad Fawwas Dzulfahmi/tugas4.py"
Mahasiswa ke-16 | Tugas: 91 | UTS: 94 | UAS: 93
Mahasiswa ke-17 | Tugas: 73 | UTS: 78 | UAS: 75
Mahasiswa ke-18 | Tugas: 87 | UTS: 84 | UAS: 89
Mahasiswa ke-19 | Tugas: 65 | UTS: 68 | UAS: 70
Mahasiswa ke-20 | Tugas: 93 | UTS: 90 | UAS: 95
Mahasiswa ke-21 | Tugas: 77 | UTS: 80 | UAS: 78
Mahasiswa ke-22 | Tugas: 82 | UTS: 84 | UAS: 88
Mahasiswa ke-23 | Tugas: 89 | UTS: 85 | UAS: 90
Mahasiswa ke-24 | Tugas: 71 | UTS: 74 | UAS: 76
```

Tugas 4.1 : output program menampilkan nilai

Untuk menampilkannya kita bisa menggunakan 1 baris kode seperti pada gambar *tugas 4.1 : source code menampilkan nilai mahasiswa*. Lalu gunakan perulangan for i in range(len(nilai)): untuk mengakses setiap baris atau data mahasiswa berdasarkan indeksinya. Fungsi len(nilai) digunakan untuk menghitung jumlah mahasiswa yang tersimpan dalam list. Di dalam perulangan, setiap nilai ditampilkan dengan perintah print(f"Mahasiswa ke-{i+1} | Tugas: {nilai[i][0]} | UTS: {nilai[i][1]} | UAS: {nilai[i][2]}"). Kode ini memanfaatkan f-string agar hasil tampilannya rapi dan mudah dibaca, menampilkan urutan mahasiswa serta nilai tugas, UTS, dan UAS yang sesuai dengan indeks elemen di dalam list. Dan hasilnya akan terlihat seperti pada gambar *Tugas 4.1 :output program menampilkan nilai*

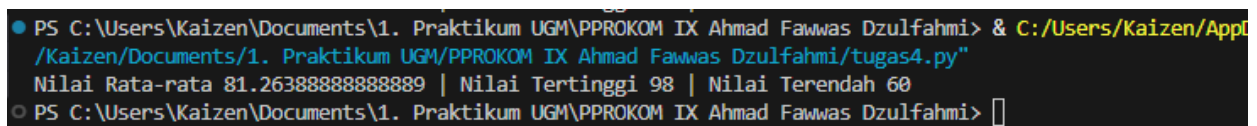


```
Tugas 4.2

nilai = np.array(nilai)
rata = np.mean(nilai)
tertinggi = np.max(nilai)
terendah = np.min(nilai)

print(f"Nilai Rata-rata {rata} | Nilai Tertinggi {tertinggi} | Nilai Terendah {terendah}")
```

Tugas 4.2 : source code program menghitung nilai rata-rata, tertinggi dan terendah



```
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi> & C:/Users/Kaizen/AppData/Local/Programs/Python/Python39-64/Python.exe C:/Users/Kaizen/AppData/Local/Programs/Python/Python39-64/Python.exe C:/Users/Kaizen/Documents/1. Praktikum UGM/PPROKOM IX Ahmad Fawwas Dzulfahmi/tugas4.py
Nilai Rata-rata 81.26388888888889 | Nilai Tertinggi 98 | Nilai Terendah 60
PS C:\Users\Kaizen\Documents\1. Praktikum UGM\PPROKOM IX Ahmad Fawwas Dzulfahmi>
```

Tugas 4.2 : output program menghitung nilai rata-rata, tertinggi dan terendah

Untuk mengolah datanya kita bisa menggunakan baris kode seperti diatas. data nilai yang sebelumnya berbentuk list dua dimensi diubah menjadi array NumPy dengan perintah nilai = np.array(nilai). Kemudian fungsi np.mean(nilai) digunakan untuk menghitung nilai rata-rata dari seluruh data dalam array, sedangkan np.max(nilai) digunakan untuk mencari nilai tertinggi, dan np.min(nilai) untuk mencari nilai terendah.

Hasil dari ketiga perhitungan tersebut disimpan ke dalam variabel rata, tertinggi, dan terendah. Pada bagian akhir, perintah print(f"Nilai Rata-rata {rata} | Nilai Tertinggi {tertinggi} | Nilai Terendah {terendah}") digunakan untuk menampilkan hasil perhitungan. Dengan cara ini, program dapat menampilkan informasi statistik dasar dari seluruh data nilai mahasiswa secara ringkas dan cepat tanpa perlu melakukan perulangan manual. Dan hasilnya akan seperti di gambar *tugas 4.2 : output program menghitung nilai rata-rata, tertinggi dan terendah*.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan hasil praktikum yang dilakukan, seluruh rumusan masalah telah berhasil diselesaikan dengan menggunakan berbagai konsep array dan operasi multidimensi pada Python. Pada rumusan masalah pertama, program berhasil menampilkan elemen-elemen tertentu dari array tiga dimensi, baik dengan menampilkan seluruh elemen pada lapisan pertama menggunakan teknik slicing maupun menampilkan elemen kolom terakhir dari setiap baris dan lapisan melalui struktur perulangan bertingkat.

Selanjutnya, pada rumusan masalah kedua, penggunaan nested list comprehension digunakan untuk membuat pola matriks identitas dengan ukuran yang dapat ditentukan secara dinamis oleh pengguna. Hal ini mempermudah pembuatan matriks karena proses pembentukan baris dan kolom dilakukan dalam satu baris kode dengan sintaks yang ringkas dan mudah dibaca.

Untuk rumusan masalah ketiga, operasi aljabar sederhana seperti penjumlahan, pengurangan, dan perkalian matriks dilakukan menggunakan library NumPy. Operasi penjumlahan dan pengurangan dilakukan secara langsung antar elemen pada posisi yang sama, sedangkan perkalian matriks dilakukan dengan metode dot() setelah matriks kedua ditranspose. Hasil yang diperoleh sesuai dengan aturan aljabar linear.

Terakhir, pada rumusan masalah keempat, program mampu menampilkan data nilai mahasiswa dalam bentuk tabel sederhana menggunakan perulangan serta menghitung nilai rata-rata, nilai tertinggi, dan nilai terendah dengan bantuan fungsi mean(), max(), dan min() dari NumPy. Yang mana ini dapat mempermudah kita dalam membaca sebuah data.

DAFTAR PUSTAKA

- Agnes. (2022, August 9). Tutorial Penggunaan Array Python untuk Pemula. Dqlab.id; DQLab | Kursus Data Science Online Indonesia R Python. <https://dqlab.id/tutorial-penggunaan-array-python-untuk-pemula>
- GeeksforGeeks. (2018, December 9). Multidimensional lists in Python. GeeksforGeeks. <https://www.geeksforgeeks.org/python/multi-dimensional-lists-in-python/>
- Memahami: Apa itu Numpy pada Python? (2022, November 12). Petani Kode. <https://www.petanikode.com/python-numpy/>
- NumPy Creating Arrays. (n.d.). Wwww.w3schools.com. https://www.w3schools.com/python/numpy/numpy_creating_arrays.asp
- Python Matriks: Contoh Transpose, Perkalian, Array NumPy. (2024, August 12). Guru99. <https://www.guru99.com/id/python-matrix.html>
- Python Matrix and Introduction to NumPy. (n.d.). Wwww.programiz.com. <https://www.programiz.com/python-programming/matrix>
- Wede. (2021, January 31). Metode Dasar Numpy Array dalam Mengolah Data. Dqlab.id; DQLab | Kursus Data Science Online Indonesia R Python. <https://dqlab.id/metode-dasar-numpy-array-dalam-mengolah-data>