

NAMA: Septia agus rinato

NIM: 24241022

KELAS: PTI A

## Percobaan 1: array

### Peraktek 1:

```
# Praktek 1 : Membuat array
# impor library numpy
import numpy as np

# membuat array dengan numpy
nilai_siswa = np.array([85, 55, 40, 90])

# akses data pada array
print(nilai_siswa[3])
```

Hasil:



90

Penjelasan:

Praktek 1 : Membuat array

impor library numpy

import numpy as np

- Baris pertama adalah komentar yang menjelaskan bahwa kode ini digunakan untuk membuat array.
- Baris kedua adalah komentar yang menjelaskan bahwa library NumPy akan diimpor.
- `import numpy as np`: Mengimpor library NumPy dan memberikan alias np untuk memudahkan penggunaan.

membuat array dengan numpy

```
nilai_siswa = np.array([85, 55, 40, 90])
```

- Baris pertama adalah komentar yang menjelaskan bahwa array akan dibuat menggunakan NumPy.
- `nilai_siswa = np.array([85, 55, 40, 90])`: Membuat array dengan menggunakan fungsi `np.array()` dan mengisi array dengan nilai-nilai siswa.

akses data pada array

```
print(nilai_siswa[3])
```

- Baris pertama adalah komentar yang menjelaskan bahwa data pada array akan diakses.
- `print(nilai_siswa[3])`: Mengakses data pada array dengan indeks 3 dan mencetak hasilnya.

## Peraktek 2:

```

# Praktek 2 : Mengakses, Mengubah, dan Cek Ukuran dan Dimensi Array
# impor library numpy
import numpy as np

# membuat array dengan numpy
nilai_siswa_1 = np.array([75, 65, 45, 80])
nilai_siswa_2 = np.array([[85, 55, 40], [50, 40, 99]])

# cara akses elemen array
print(nilai_siswa_1[0])
print(nilai_siswa_2[1][1])

# mengubah nilai elemen array
nilai_siswa_1[0] = 88
nilai_siswa_2[1][1] = 70

# cek perubahannya
print(nilai_siswa_1[0])
print(nilai_siswa_2[1][1])

# Cek ukuran dan dimensi array
print("Ukuran Array : ", nilai_siswa_1.shape)
print("Ukuran Array : ", nilai_siswa_2.shape)
print("Dimensi Array : ", nilai_siswa_2.ndim)

```

**Hasil:**

```

75
40
88
70
Ukuran Array :  (4,)
Ukuran Array :  (2, 3)
Dimensi Array :  2

```

**Penjelasan:**

**Mengimpor Library NumPy**

- Library NumPy diimpor untuk digunakan dalam membuat dan mengolah array.

**Membuat Array**

- Dua array dibuat menggunakan library NumPy, yaitu `nilai_siswa_1` dan `nilai_siswa_2`.
- `nilai_siswa_1` adalah array 1 dimensi dengan 4 elemen, yaitu `[75, 65, 45, 80]`.
- `nilai_siswa_2` adalah array 2 dimensi dengan 2 baris dan 3 kolom, yaitu `[[85, 55, 40], [50, 40, 99]]`.

### Mengakses Elemen Array

- Elemen array dapat diakses menggunakan indeks.
- `nilai_siswa_1[0]` mengakses elemen pertama dari array 1 dimensi, yaitu 75.
- `nilai_siswa_2[1][1]` mengakses elemen baris ke-2, kolom ke-2 dari array 2 dimensi, yaitu 40.

### Mengubah Nilai Elemen Array

- Nilai elemen array dapat diubah menggunakan indeks.
- `nilai_siswa_1[0] = 88` mengubah nilai elemen pertama dari array 1 dimensi menjadi 88.
- `nilai_siswa_2[1][1] = 70` mengubah nilai elemen baris ke-2, kolom ke-2 dari array 2 dimensi menjadi 70.

### Cek Ukuran dan Dimensi Array

- Ukuran array dapat dicek menggunakan atribut `shape`.
- Dimensi array dapat dicek menggunakan atribut `ndim`.
- `nilai_siswa_1.shape` mengembalikan ukuran array 1 dimensi, yaitu `(4,)`.
- `nilai_siswa_2.shape` mengembalikan ukuran array 2 dimensi, yaitu `(2, 3)`.
- `nilai_siswa_2.ndim` mengembalikan dimensi array 2 dimensi, yaitu 2.

Dengan demikian, kode program ini mendemonstrasikan cara membuat array, mengakses elemen array, mengubah nilai elemen array, dan cek ukuran dan dimensi array menggunakan library NumPy.

## Peraktek 3:

```
# Praktek 3 : Operasi Aritmatika Pada Array
# impor library numpy
import numpy as np

# membuat array
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])

# menggunakan operasi penjumlahan pada 2 array
print(a + b)          # array([5, 7, 9])

# Indexing dan Slicing pada Array
arr = np.array([10, 20, 30, 40])
print(arr[1:3])       # array([20, 30])

# iterasi pada array
for x in arr:
    print(x)
```

Hasil:

```
[5 7 9]
[20 30]
10
20
30
40
```

Penjelasan:

1. `import numpy as np`:

- Baris ini digunakan untuk mengimpor library NumPy dan memberikan alias np untuk memudahkan penggunaan.

2. `a = np.array([1, 2, 3])`:

- Baris ini digunakan untuk membuat array a dengan elemen [1, 2, 3].

**3. `b = np.array([4, 5, 6])`:**

- Baris ini digunakan untuk membuat array `b` dengan elemen `[4, 5, 6]`.

**4. `print(a + b)`:**

- Baris ini digunakan untuk melakukan operasi penjumlahan pada array `a` dan `b`.

- Hasil penjumlahan adalah array baru dengan elemen `[5, 7, 9]`.

**5. `arr = np.array([10, 20, 30, 40])`:**

- Baris ini digunakan untuk membuat array `arr` dengan elemen `[10, 20, 30, 40]`.

**6. `print(arr[1:3])`:**

- Baris ini digunakan untuk mengakses elemen array `arr` dari indeks 1 sampai 2 (indeks 3 tidak termasuk).

- Hasilnya adalah array dengan elemen `[20, 30]`.

**7. `for x in arr::`**

- Baris ini digunakan untuk melakukan iterasi pada array `arr`.

- Setiap elemen array `arr` diakses dan dicetak satu per satu.

**8. `print(x)`:**

- Baris ini digunakan untuk mencetak setiap elemen array `arr` yang diakses dalam iterasi.

## **Percobaan 2: Traversal, Insertion, Deletion**

### **Peraktek 4:**

```
# Praktek 4 : Linear Traversal
# membuat array
arr = [1, 2, 3, 4, 5]

# Linear Traversal ke tiap elemen arr
print("Linear Traversal: ", end=" ")
for i in arr:
    print(i, end=" ")
print()
```

**Hasil:**

```
Linear Traversal:  1 2 3 4 5
```

**Penjelasan:**

**1. arr = [1, 2, 3, 4, 5]:**

- Baris ini digunakan untuk membuat array arr dengan elemen [1, 2, 3, 4, 5].

**2. print("Linear Traversal: ", end=" "):**

- Baris ini digunakan untuk mencetak string "Linear Traversal: " dan tidak membuat baris baru setelah mencetak string tersebut.

**3. for i in arr::**

- Baris ini digunakan untuk melakukan iterasi pada array arr.

- Setiap elemen array arr diakses dan dicetak satu per satu.

**4. print(i, end=" "):**

- Baris ini digunakan untuk mencetak setiap elemen array arr yang diakses dalam iterasi.

- end=" " digunakan untuk mencetak elemen-elemen array dalam satu baris dengan dipisahkan oleh spasi.

## 5. print():

- Baris ini digunakan untuk membuat baris baru setelah mencetak semua elemen array.

## Peraktek 5:

```
# Praktek 5 : Reverse Traversal
# membuat array
arr = [1, 2, 3, 4, 5]

# Reverse Traversal dari elemen akhir
print("Reverse Traversal: ", end="")
for i in range(len(arr) - 1, -1, -1):
    print(arr[i], end=" ")
print()
```

### Hasil:

```
Reverse Traversal: 5 4 3 2 1
```

### Penjelasan:

#### 1. arr = [1, 2, 3, 4, 5]:

- Baris ini digunakan untuk membuat array arr dengan elemen [1, 2, 3, 4, 5].

#### 2. print("Reverse Traversal: ", end=""):

- Baris ini digunakan untuk mencetak string "Reverse Traversal: " dan tidak membuat baris baru setelah mencetak string tersebut.

#### 3. for i in range(len(arr) - 1, -1, -1)::

- Baris ini digunakan untuk melakukan iterasi pada array arr dari elemen terakhir ke elemen pertama.
- len(arr) - 1 digunakan untuk mendapatkan indeks elemen terakhir array.
- -1 digunakan sebagai batas akhir iterasi (indeks sebelum elemen pertama).



- -1 sebagai step digunakan untuk mengurangi indeks pada setiap iterasi.

4. `print(arr[i], end=" ")`:

- Baris ini digunakan untuk mencetak setiap elemen array `arr` yang diakses dalam iterasi.

- `end=" "` digunakan untuk mencetak elemen-elemen array dalam satu baris dengan dipisahkan oleh spasi.

5. `print()`:

- Baris ini digunakan untuk membuat baris baru setelah mencetak semua elemen array.

## Peraktek 7:

```
# Praktek 7 : Linear Traversal dengan Metode While
# membuat array
arr = [1, 2, 3, 4, 5]

# mendeklarasikan nilai awal
n = len(arr)
i = 0

print("Linear Traversal using while loop: ", end=" ")
# Linear Traversal dengan while
while i < n:
    print(arr[i], end=" ")
    i += 1
print()
```

Hasil:

```
Traversal using while loop: 1 2 3 4 5
```

Penjelasan:

1. `arr = [1, 2, 3, 4, 5]`:

- Baris ini digunakan untuk membuat array `arr` dengan elemen `[1, 2, 3, 4, 5]`.

**2. `n = len(arr):`**

- Baris ini digunakan untuk mendapatkan panjang array `arr` dan menyimpannya dalam variabel `n`.

**3. `i = 0:`**

- Baris ini digunakan untuk mendeklarasikan variabel `i` sebagai indeks awal array, yaitu 0.

**4. `print("Linear Traversal using while loop: ", end=" "):`**

- Baris ini digunakan untuk mencetak string "Linear Traversal using while loop: " dan tidak membuat baris baru setelah mencetak string tersebut.

**5. `while i < n::`**

- Baris ini digunakan untuk melakukan perulangan `while` selama kondisi `i < n` terpenuhi.

- Perulangan akan berhenti ketika `i` mencapai nilai `n`, yaitu panjang array.

**6. `print(arr[i], end=" "):`**

- Baris ini digunakan untuk mencetak setiap elemen array `arr` yang diakses dalam perulangan.

- `end=" "` digunakan untuk mencetak elemen-elemen array dalam satu baris dengan dipisahkan oleh spasi.

**7. `i += 1:`**

- Baris ini digunakan untuk meningkatkan nilai `i` sebanyak 1 pada setiap iterasi perulangan.

**8. `print():`**

- Baris ini digunakan untuk membuat baris baru setelah mencetak semua elemen array.

## Peraktek 8:

```
# Praktek 8 : Reverse Traversal dengan Metode While
# membuat array
arr = [1, 2, 3, 4, 5]

# mendeklarasikan nilai awal
start = 0
end = len(arr) - 1

print("Reverse Traversal using while loop: ", end=" ")
# Reverse Traversal dengan while
while start < end:
    # mengubah indeks array
    arr[start], arr[end] = arr[end], arr[start]
    start += 1
    end -= 1
print(arr)
```

### Hasil:

```
Reverse Traversal using while loop: [5, 4, 3, 2, 1]
In [18]:
```

### Penjelasan:

1. `arr = [1, 2, 3, 4, 5]`:

- Baris ini membuat sebuah array `arr` dengan 5 elemen, yaitu 1, 2, 3, 4, dan 5.

2. `start = 0`:

- Baris ini mendeklarasikan variabel `start` sebagai indeks awal array, yaitu 0.

3. `end = len(arr) - 1`:

- `len(arr)` mendapatkan panjang array `arr`, yaitu 5.
- `len(arr) - 1` mendapatkan indeks terakhir array, yaitu 4.
- Baris ini mendeklarasikan variabel `end` sebagai indeks terakhir array.

4. `print("Reverse Traversal using while loop: ", end=" ")`:

- Baris ini mencetak string "Reverse Traversal using while loop: " tanpa membuat baris baru setelahnya.

5. `while start < end::`

- Baris ini melakukan perulangan `while` selama kondisi `start < end` terpenuhi.
- Perulangan akan berhenti ketika `start` mencapai atau melebihi nilai `end`.

6. `arr[start], arr[end] = arr[end], arr[start]`:

- Baris ini menukar nilai elemen array pada indeks `start` dan `end`.
- Dengan cara ini, elemen-elemen array dapat dibalik urutannya.

7. `start += 1`:

- Baris ini meningkatkan nilai `start` sebanyak 1 pada setiap iterasi perulangan.

8. `end -= 1`:

- Baris ini mengurangi nilai `end` sebanyak 1 pada setiap iterasi perulangan.

9. `print(arr)`:

- Baris ini mencetak array `arr` setelah perulangan `while` selesai.
- Karena perulangan `while` telah membalik urutan elemen-elemen array, maka array yang dicetak akan memiliki urutan terbalik.

## Peraktek 9:

```

# Praktek 9 : Insertion pada akhir elemen array
# membuat array
arr = [12, 16, 20, 40, 50, 70]

# cetak arr sebelum penyisipan
print("Array Sebelum Insertion : ", arr)

# cetak panjang array sebelum penyisipan
print("Panjang Array : ", len(arr))

# menyisipkan array di akhir elemen menggunakan .append()
arr.append(26)

# cetak arr setelah penyisipan
print("Array Setelah Insertion : ", arr)

# cetak panjang array setelah penyisipan
print("Panjang Array : ", len(arr))

```

#### Hasil:

```

Array Sebelum Insertion : [12, 16, 20, 40, 50, 70]
Panjang Array : 6
Array Setelah Insertion : [12, 16, 20, 40, 50, 70, 26]
Panjang Array : 7

```

#### Penjelasan:

##### 1. # Praktek 9 : Insertion pada akhir elemen array:

- Baris ini adalah komentar yang menjelaskan tujuan kode program ini.

##### 2. # membuat array:

- Baris ini adalah komentar yang menjelaskan bahwa kode program ini akan membuat sebuah array.

##### 3. arr = [12, 16, 20, 40, 50, 70]:

- Baris ini membuat sebuah array arr dengan 6 elemen, yaitu 12, 16, 20, 40, 50, dan 70.

##### 4. # cetak arr sebelum penyisipan:

- Baris ini adalah komentar yang menjelaskan bahwa kode program ini akan mencetak array sebelum penyisipan.

5. `print("Array Sebelum Insertion : ", arr):`

- Baris ini mencetak string "Array Sebelum Insertion : " diikuti oleh isi array arr sebelum penyisipan.

6. `# cetak panjang array sebelum penyisipan:`

- Baris ini adalah komentar yang menjelaskan bahwa kode program ini akan mencetak panjang array sebelum penyisipan.

7. `print("Panjang Array : ", len(arr)):`

- `len(arr)` mendapatkan panjang array arr, yaitu 6.

- Baris ini mencetak string "Panjang Array : " diikuti oleh panjang array arr sebelum penyisipan.

8. `# menyisipkan array di akhir elemen menggunakan .append():`

- Baris ini adalah komentar yang menjelaskan bahwa kode program ini akan menyisipkan elemen di akhir array menggunakan metode `.append()`.

9. `arr.append(26):`

- `.append()` adalah metode yang digunakan untuk menyisipkan elemen di akhir array.

- Baris ini menyisipkan elemen 26 di akhir array arr.

10. `# cetak arr setelah penyisipan:`

- Baris ini adalah komentar yang menjelaskan bahwa kode program ini akan mencetak array setelah penyisipan.

11. `print("Array Setelah Insertion : ", arr):`

- Baris ini mencetak string "Array Setelah Insertion : " diikuti oleh isi array arr setelah penyisipan.

12. # cetak panjang array setelah penyisipan:

- Baris ini adalah komentar yang menjelaskan bahwa kode program ini akan mencetak panjang array setelah penyisipan.

13. print("Panjang Array : ", len(arr)):

- len(arr) mendapatkan panjang array arr setelah penyisipan, yaitu 7.

- Baris ini mencetak string "Panjang Array : " diikuti oleh panjang array arr setelah penyisipan.

## Peraktek 10:

```
# Praktek 10 : Insertion pada tengah elemen array
# membuat array
arr = [12, 16, 20, 40, 50, 70]

# cetak arr sebelum penyisipan
print("Array Sebelum Insertion : ", arr)

# cetak panjang array sebelum penyisipan
print("Panjang Array : ", len(arr))

# menyisipkan array pada tengah elemen menggunakan .insert(pos, x)
arr.insert(4, 5)

# cetak arr setelah penyisipan
print("Array Setelah Insertion : ", arr)

# cetak panjang array setelah penyisipan
print("Panjang Array : ", len(arr))
```

Hasil:

```
Array Sebelum Insertion : [12, 16, 20, 40, 50, 70]
Panjang Array : 6
Array Setelah Insertion : [12, 16, 20, 40, 5, 50, 70]
Panjang Array : 7
```

Penjelasan:

**1. arr = [12, 16, 20, 40, 50, 70]:**

- Baris ini membuat sebuah array arr dengan 6 elemen, yaitu 12, 16, 20, 40, 50, dan 70.

**2. print("Array Sebelum Insertion : ", arr):**

- Baris ini mencetak string "Array Sebelum Insertion : " diikuti oleh isi array arr sebelum penyisipan.

**3. print("Panjang Array : ", len(arr)):**

- len(arr) mendapatkan panjang array arr, yaitu 6.

- Baris ini mencetak string "Panjang Array : " diikuti oleh panjang array arr sebelum penyisipan.

**4. arr.insert(4, 5):**

- .insert() adalah metode yang digunakan untuk menyisipkan elemen pada posisi tertentu dalam array.

- 4 adalah indeks posisi penyisipan, dan 5 adalah nilai elemen yang disisipkan.

- Baris ini menyisipkan elemen 5 pada indeks ke-4 dalam array arr.

**5. print("Array Setelah Insertion : ", arr):**

- Baris ini mencetak string "Array Setelah Insertion : " diikuti oleh isi array arr setelah penyisipan.

**6. print("Panjang Array : ", len(arr)):**

- len(arr) mendapatkan panjang array arr setelah penyisipan, yaitu 7.

- Baris ini mencetak string "Panjang Array : " diikuti oleh panjang array arr setelah penyisipan.



```

# jika tidak menggunakan fungsi .insert()
# membuat array dan cetak array
arr = [12, 16, 20, 40, 50, 70]
print("Array Sebelum Penyisipan : " , arr)

# Deklarasi elemen tengah yang disisipkan
pos = 4

# Deklarasi nilai yang akan disisipkan
x = 5

# menambah elemen dummy agar menambah panjang array
arr.append(0) # arr = [12, 16, 20, 40, 50, 70, 0]

# melakukan pergeseran elemen mulai dari belakang
for i in range(len(arr) - 2, pos-1, -1):
    arr[i + 1] = arr[i]

# memasukkan nilai x pada elemen yang diinginkan
arr[pos] = x

# Cetak array baru
print("Array Sesudah Penyisipan : ", arr)

```

**Hasil:**

```

Array Sebelum Penyisipan : [12, 16, 20, 40, 50, 70]
Array Sesudah Penyisipan : [12, 16, 20, 40, 5, 50, 70]

```

**Penjelasan:**

**1. # Praktek 10 : Insertion pada tengah elemen array:**

- Baris ini adalah komentar yang menjelaskan tujuan kode program ini.

**2. arr = [12, 16, 20, 40, 50, 70]:**

- Baris ini membuat sebuah array arr dengan 6 elemen, yaitu 12, 16, 20, 40, 50, dan 70.

**3. print("Array Sebelum Insertion : ", arr):**

- Baris ini mencetak string "Array Sebelum Insertion : " diikuti oleh isi array arr sebelum penyisipan.

**4. print("Panjang Array : ", len(arr)):**

- len(arr) mendapatkan panjang array arr, yaitu 6.
- Baris ini mencetak string "Panjang Array : " diikuti oleh panjang array arr sebelum penyisipan.

**5. arr.insert(4, 5):**

- .insert() adalah metode yang digunakan untuk menyisipkan elemen pada posisi tertentu dalam array.
- 4 adalah indeks posisi penyisipan, dan 5 adalah nilai elemen yang disisipkan.
- Baris ini menyisipkan elemen 5 pada indeks ke-4 dalam array arr.

**6. print("Array Setelah Insertion : ", arr):**

- Baris ini mencetak string "Array Setelah Insertion : " diikuti oleh isi array arr setelah penyisipan.

**7. print("Panjang Array : ", len(arr)):**

- len(arr) mendapatkan panjang array arr setelah penyisipan, yaitu 7.
- Baris ini mencetak string "Panjang Array : " diikuti oleh panjang array arr setelah penyisipan.

## **Peraktek 11:**

```

# Praktek 11 : Menghapus array
# membuat array
a = [10, 20, 30, 40, 50]
print("Array Sebelum Deletion : ", a)

# menghapus elemen array pertama yang nilainya 30
a.remove(30)
print("Setelah remove(30):", a)

# menghapus elemen array pada index 1 (20)
popped_val = a.pop(1)
print("Popped element:", popped_val)
print("Setelah pop(1):", a)

# Menghapus elemen pertama (10)
del a[0]
print("Setelah del a[0]:", a)

```

#### Hasil:

```

Array Sebelum Deletion : [10, 20, 30, 40, 50]
Setelah remove(30): [10, 20, 40, 50]
Popped element: 20
Setelah pop(1): [10, 40, 50]
Setelah del a[0]: [40, 50]

```

#### Penjelasan:

##### 1. a = [10, 20, 30, 40, 50]:

- Baris ini membuat sebuah array a dengan 5 elemen, yaitu 10, 20, 30, 40, dan 50.

##### 2. print("Array Sebelum Deletion : ", a):

- Baris ini mencetak string "Array Sebelum Deletion : " diikuti oleh isi array a sebelum penghapusan.

##### 3. a.remove(30):

- .remove() adalah metode yang digunakan untuk menghapus elemen pertama yang nilainya sesuai dengan argumen.

- Baris ini menghapus elemen 30 dari array a.

4. `print("Setelah remove(30):", a):`

- Baris ini mencetak string "Setelah remove(30):" diikuti oleh isi array a setelah penghapusan elemen 30.

5. `popped_val = a.pop(1):`

- `.pop()` adalah metode yang digunakan untuk menghapus elemen pada indeks tertentu dan mengembalikan nilai elemen tersebut.

- 1 adalah indeks elemen yang akan dihapus, yaitu elemen kedua (20).

- Baris ini menghapus elemen 20 dari array a dan mengembalikan nilai 20 ke variabel `popped_val`.

6. `print("Popped element:", popped_val):`

- Baris ini mencetak string "Popped element:" diikuti oleh nilai elemen yang dihapus (20).

7. `print("Setelah pop(1):", a):`

- Baris ini mencetak string "Setelah pop(1):" diikuti oleh isi array a setelah penghapusan elemen 20.

8. `del a[0]:`

- `del` adalah pernyataan yang digunakan untuk menghapus elemen pada indeks tertentu.

- 0 adalah indeks elemen yang akan dihapus, yaitu elemen pertama (10).

- Baris ini menghapus elemen 10 dari array a.

9. `print("Setelah del a[0]:", a):`

- Baris ini mencetak string "Setelah del a[0]:" diikuti oleh isi array a setelah penghapusan elemen 10.

## Percobaan 3: Matriks atau Array 2 dan 3 Dimensi

### Peraktek 12:

```
# Praktek 12 : Membuat Matriks 2 Dimensi
# impor library numpy
import numpy as np

# membuat matiks dengan numpy
matriks_np = np.array([[1,2,3],
                       [4,5,6],
                       [7,8,9]])

print(matriks_np[2][2])
```

Hasil:

9

Penjelasan:

1. import numpy as np:

- Baris ini mengimpor library numpy dan memberikan alias np untuk memudahkan penggunaan.
- Numpy adalah library Python yang digunakan untuk operasi numerik dan matriks.

2. matriks\_np = np.array([[1,2,3], [4,5,6], [7,8,9]]):

- Baris ini membuat sebuah matriks 2 dimensi menggunakan numpy.
- np.array() adalah fungsi yang digunakan untuk membuat array numpy.
- [[1,2,3], [4,5,6], [7,8,9]] adalah data yang digunakan untuk membuat matriks.

- Matriks ini memiliki 3 baris dan 3 kolom.

3. `print(matriks_np[2][2]):`

- Baris ini mencetak elemen matriks pada indeks baris 2 dan indeks kolom 2.
- Dalam Python, indeks dimulai dari 0, sehingga indeks 2 sebenarnya adalah elemen ketiga.
- Elemen pada indeks baris 2 dan indeks kolom 2 adalah 9.

Matriks yang dibuat adalah:

`[[1 2 3]`

`[4 5 6]`

`[7 8 9]]`

## Peraktek 13:

```

# Praktek 13 : Operasi Penjumlahan Matriks dengan List
# Program penjumlahan matriks yang dibuat dari list

X = [[12,7,3],
      [4,5,6],
      [7,8,9]]

Y = [[5,8,1],
      [6,7,3],
      [4,5,9]]

result = [[0,0,0],
          [0,0,0],
          [0,0,0]]

# proses penjumlahan dua matriks menggunakan nested loop
# mengulang sebanyak row (baris)
for i in range(len(X)):
    # mengulang sebanyak column (kolom)
    for j in range(len(X[0])):
        result[i][j] = X[i][j] + Y[i][j]

print("Hasil Penjumlahan Matriks dari LIST")

# cetak hasil penjumlahan secara iteratif
for r in result:
    print(r)

```

**Hasil:**

```

Hasil Penjumlahan Matriks dari LIST
[17, 15, 4]
[10, 12, 9]
[11, 13, 18]

```

**Penjelasan:**

1.  $X = [[12,7,3], [4,5,6], [7,8,9]]$ :

- Baris ini membuat sebuah matriks 2 dimensi X dengan 3 baris dan 3 kolom.

2.  $Y = [[5,8,1], [6,7,3], [4,5,9]]$ :

- Baris ini membuat sebuah matriks 2 dimensi Y dengan 3 baris dan 3 kolom.

**3. result = [[0,0,0], [0,0,0], [0,0,0]]:**

- Baris ini membuat sebuah matriks 2 dimensi result dengan 3 baris dan 3 kolom, yang akan digunakan untuk menyimpan hasil penjumlahan matriks X dan Y.

**4. for i in range(len(X))::**

- Baris ini melakukan perulangan sebanyak jumlah baris matriks X.

- len(X) mengembalikan jumlah baris matriks X, yaitu 3.

**5. for j in range(len(X[0]))::**

- Baris ini melakukan perulangan sebanyak jumlah kolom matriks X.

- len(X[0]) mengembalikan jumlah kolom matriks X, yaitu 3.

**6. result[i][j] = X[i][j] + Y[i][j]:**

- Baris ini menambahkan elemen matriks X dan Y pada indeks baris i dan indeks kolom j, dan menyimpan hasilnya pada matriks result.

**7. print("Hasil Penjumlahan Matriks dari LIST"):**

- Baris ini mencetak string "Hasil Penjumlahan Matriks dari LIST".

**8. for r in result::**

- Baris ini melakukan perulangan sebanyak jumlah baris matriks result.

- r adalah variabel yang digunakan untuk mengakses setiap baris matriks result.

**9. print(r):**

- Baris ini mencetak setiap baris matriks result.

**Proses penjumlahan matriks:**

- Matriks X adalah [[12, 7, 3], [4, 5, 6], [7, 8, 9]].

- Matriks Y adalah [[5, 8, 1], [6, 7, 3], [4, 5, 9]].



- Hasil penjumlahan matriks X dan Y adalah  $\begin{bmatrix} 17 & 15 & 4 \\ 10 & 12 & 9 \\ 11 & 13 & 18 \end{bmatrix}$ .

## Praktek 14:

```
# Praktek 14 : Operasi Penjumlahan Matriks dengan numpy
# impor library numpy
import numpy as np

# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

Y = np.array([
    [5,8,1],
    [6,7,3],
    [4,5,9]])

# Operasi penjumlahan dua matrik numpy
result = X + Y

# cetak hasil
print("Hasil Penjumlahan Matriks dari NumPy")
print(result)
```

Hasil:

```
Hasil Penjumlahan Matriks dari NumPy
[[17 15  4]
 [10 12  9]
 [11 13 18]]
```

Penjelasan:

1. import numpy as np:

- Baris ini mengimpor library numpy dan memberikan alias np untuk memudahkan penggunaan.

- Numpy adalah library Python yang digunakan untuk operasi numerik dan matriks.

2. `X = np.array([[12,7,3], [4,5,6], [7,8,9]]):`

- Baris ini membuat sebuah matriks 2 dimensi X dengan 3 baris dan 3 kolom menggunakan numpy.

- `np.array()` adalah fungsi yang digunakan untuk membuat array numpy.

3. `Y = np.array([[5,8,1], [6,7,3], [4,5,9]]):`

- Baris ini membuat sebuah matriks 2 dimensi Y dengan 3 baris dan 3 kolom menggunakan numpy.

4. `result = X + Y:`

- Baris ini melakukan operasi penjumlahan matriks X dan Y menggunakan operator +.

- Numpy secara otomatis melakukan penjumlahan elemen-wise antara dua matriks.

5. `print("Hasil Penjumlahan Matriks dari NumPy"):`

- Baris ini mencetak string "Hasil Penjumlahan Matriks dari NumPy".

6. `print(result):`

- Baris ini mencetak hasil penjumlahan matriks X dan Y.

Proses penjumlahan matriks:

- Matriks X adalah `[[12, 7, 3], [4, 5, 6], [7, 8, 9]]`.

- Matriks Y adalah `[[5, 8, 1], [6, 7, 3], [4, 5, 9]]`.

- Hasil penjumlahan matriks X dan Y adalah `[[17, 15, 4], [10, 12, 9], [11, 13, 18]]`.

## Peraktek 15:

```
# Praktek 15 : Operasi Pengurangan Matriks dengan numpy
# impor library numpy
import numpy as np

# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

Y = np.array([
    [5,8,1],
    [6,7,3],
    [4,5,9]])

# Operasi pengurangan dua matrik numpy
result = X - Y

# cetak hasil
print("Hasil Pengurangan Matriks dari NumPy")
print(result)
```

**Hasil:**

```
Hasil Pengurangan Matriks dari LIST
[[ 7 -1  2]
 [-2 -2  3]
 [ 3  3  0]]
```

**Penjelasan:**

**1. import numpy as np:**

- Baris ini mengimpor library numpy dan memberikan alias np untuk memudahkan penggunaan.
- Numpy adalah library Python yang digunakan untuk operasi numerik dan matriks.

**2. X = np.array([[12,7,3], [4,5,6], [7,8,9]]):**

- Baris ini membuat sebuah matriks 2 dimensi X dengan 3 baris dan 3 kolom menggunakan numpy.

- np.array() adalah fungsi yang digunakan untuk membuat array numpy.

3. Y = np.array([[5,8,1], [6,7,3], [4,5,9]]):

- Baris ini membuat sebuah matriks 2 dimensi Y dengan 3 baris dan 3 kolom menggunakan numpy.

4. result = X - Y:

- Baris ini melakukan operasi pengurangan matriks X dan Y menggunakan operator -.

- Numpy secara otomatis melakukan pengurangan elemen-wise antara dua matriks.

5. print("Hasil Pengurangan Matriks dari NumPy"):

- Baris ini mencetak string "Hasil Pengurangan Matriks dari NumPy".

6. print(result):

- Baris ini mencetak hasil pengurangan matriks X dan Y.

## Praktek 16:

```
# Praktek 16 : Operasi Perkalian Matriks dengan numpy
# impor library numpy
import numpy as np

# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

Y = np.array(
    [[5,8,1],
    [6,7,3],
    [4,5,9]])

# Operasi perkalian dua matrik numpy
result = X * Y

# cetak hasil
print("Hasil Perkalian Matriks dari NumPy")
print(result)
```

## Hasil:

```
Hasil Perkalian Matriks dari LIST
[[60 56  3]
 [24 35 18]
 [28 40 81]]
```

## Penjelasan:

### 1. import numpy as np:

- Baris ini mengimpor library numpy dan memberikan alias np untuk memudahkan penggunaan.
- Numpy adalah library Python yang digunakan untuk operasi numerik dan matriks.

### 2. X = np.array([[12,7,3], [4,5,6], [7,8,9]]):

- Baris ini membuat sebuah matriks 2 dimensi X dengan 3 baris dan 3 kolom menggunakan numpy.
- np.array() adalah fungsi yang digunakan untuk membuat array numpy.

### 3. Y = np.array([[5,8,1], [6,7,3], [4,5,9]]):

- Baris ini membuat sebuah matriks 2 dimensi Y dengan 3 baris dan 3 kolom menggunakan numpy.

### 4. result = X \* Y:

- Baris ini melakukan operasi perkalian elemen-wise antara matriks X dan Y menggunakan operator \*.
- Perlu diingat bahwa ini bukan perkalian matriks yang sebenarnya, melainkan perkalian elemen-wise.
- Jika Anda ingin melakukan perkalian matriks yang sebenarnya, Anda dapat menggunakan np.dot(X, Y) atau X @ Y.

5. `print("Hasil Perkalian Matriks dari NumPy"):`

- Baris ini mencetak string "Hasil Perkalian Matriks dari NumPy".

6. `print(result):`

- Baris ini mencetak hasil perkalian elemen-wise antara matriks X dan Y.

## Peraktek 17:

```
# Praktek 17 : Operasi Pembagian Matriks dengan numpy
# impor library numpy
import numpy as np

# Membuat matriks dengan numpy
X = np.array([
    [12,7,3],
    [4,5,6],
    [7,8,9]])

Y = np.array([
    [5,8,1],
    [6,7,3],
    [4,5,9]])

# Operasi pembagian dua matrik numpy
result = X / Y

# cetak hasil
print("Hasil Pembagian Matriks dari NumPy")
print(result)
```

Hasil:

```
Hasil Pembagian Matriks dari LIST
[[2.4      0.875    3.      ]
 [0.66666667 0.71428571 2.      ]
 [1.75     1.6      1.      ]]
```

#### **Penjelasan:**

##### **1. import numpy as np:**

- Baris ini mengimpor library numpy dan memberikan alias np untuk memudahkan penggunaan.
- Numpy adalah library Python yang digunakan untuk operasi numerik dan matriks.

##### **2. X = np.array([[12,7,3], [4,5,6], [7,8,9]]):**

- Baris ini membuat sebuah matriks 2 dimensi X dengan 3 baris dan 3 kolom menggunakan numpy.
- np.array() adalah fungsi yang digunakan untuk membuat array numpy.

##### **3. Y = np.array([[5,8,1], [6,7,3], [4,5,9]]):**

- Baris ini membuat sebuah matriks 2 dimensi Y dengan 3 baris dan 3 kolom menggunakan numpy.

##### **4. result = X / Y:**

- Baris ini melakukan operasi pembagian elemen-wise antara matriks X dan Y menggunakan operator /.
- Perlu diingat bahwa jika ada elemen pada matriks Y yang bernilai 0, maka akan terjadi kesalahan pembagian oleh nol.

##### **5. print("Hasil Pembagian Matriks dari NumPy"):**

- Baris ini mencetak string "Hasil Pembagian Matriks dari NumPy".

##### **6. print(result):**

- Baris ini mencetak hasil pembagian elemen-wise antara matriks X dan Y.

## Praktek 18:

```
# Praktek 18 : transpose()
# impor library numpy
import numpy as np

# membuat matriks
matriks_a = np.array([
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
])

# cetak matriks
print("Matriks Sebelum Transpose")
print(matriks_a)

# transpose matriks_a
balik = matriks_a.transpose()

# cetak matriks setelah dibalik
print("Matriks Setelah Transpose")
print(balik)
```

Hasil:

```
Matriks Sebelum Transpose
[[1 2 3]
 [4 5 6]
 [7 8 9]]
Matriks Setelah Transpose
[[1 4 7]
 [2 5 8]
 [3 6 9]]
```

Penjelasan:

1. import numpy as np:

- Baris ini mengimpor library numpy dan memberikan alias np untuk memudahkan penggunaan.



- Numpy adalah library Python yang digunakan untuk operasi numerik dan matriks.

2. `matriks_a = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])`:

- Baris ini membuat sebuah matriks 2 dimensi `matriks_a` dengan 3 baris dan 3 kolom menggunakan numpy.

- `np.array()` adalah fungsi yang digunakan untuk membuat array numpy.

3. `print("Matriks Sebelum Transpose")`:

- Baris ini mencetak string "Matriks Sebelum Transpose".

4. `print(matriks_a)`:

- Baris ini mencetak matriks `matriks_a` sebelum transpose.

5. `balik = matriks_a.transpose()`:

- Baris ini melakukan operasi transpose pada matriks `matriks_a` menggunakan metode `transpose()`.

- Transpose matriks adalah operasi yang mengubah baris menjadi kolom dan kolom menjadi baris.

6. `print("Matriks Setelah Transpose")`:

- Baris ini mencetak string "Matriks Setelah Transpose".

7. `print(balik)`:

- Baris ini mencetak matriks `balik` setelah transpose

## Praktek 19:

```

# Praktek 19 : reshape()
# impor library numpy
import numpy as np

# membuat array 1 dimensi
arr_1d = np.array([50, 70, 89, 99, 103, 35])

# cetak matriks sebelum reshape
print("Matriks Sebelum Reshape")
print(arr_1d)
print("Ukuran Matriks : ", arr_1d.shape)
print("\n")

# mengubah matriks menjadi ordo 3 x 2
ubah = arr_1d.reshape(3, 2)

# cetak matriks setelah reshape ke ordo 3 x 2
print("Matriks Setelah Reshape")
print(ubah)
print("Ukuran Matriks : ", ubah.shape)

```

#### Hasil:

```

Matriks Sebelum Reshape
[ 50  70  89  99 103  35]
Ukuran Matriks : (6,)

Matriks Setelah Reshape
[[ 50  70]
 [ 89  99]
 [103  35]]
Ukuran Matriks : (3, 2)

```

#### Penjelasan:

##### 1. import numpy as np:

- Baris ini mengimpor library numpy dan memberikan alias np untuk memudahkan penggunaan.
- Numpy adalah library Python yang digunakan untuk operasi numerik dan matriks.

##### 2. arr\_1d = np.array([50, 70, 89, 99, 103, 35]):

- Baris ini membuat sebuah array 1 dimensi `arr_1d` dengan 6 elemen menggunakan `numpy`.

- `np.array()` adalah fungsi yang digunakan untuk membuat array `numpy`.

3. `print("Matriks Sebelum Reshape"):`

- Baris ini mencetak string "Matriks Sebelum Reshape".

4. `print(arr_1d):`

- Baris ini mencetak array `arr_1d` sebelum reshape.

5. `print("Ukuran Matriks : ", arr_1d.shape):`

- Baris ini mencetak ukuran matriks `arr_1d` sebelum reshape.

- Ukuran matriks `arr_1d` adalah (6,), yang berarti array 1 dimensi dengan 6 elemen.

6. `ubah = arr_1d.reshape(3, 2):`

- Baris ini mengubah array `arr_1d` menjadi matriks 2 dimensi dengan ukuran 3x2 menggunakan metode `reshape()`.

- Metode `reshape()` mengubah bentuk array tanpa mengubah data yang ada di dalamnya.

7. `print("Matriks Setelah Reshape"):`

- Baris ini mencetak string "Matriks Setelah Reshape".

8. `print(ubah):`

- Baris ini mencetak matriks `ubah` setelah reshape.

9. `print("Ukuran Matriks : ", ubah.shape):`

- Baris ini mencetak ukuran matriks `ubah` setelah reshape.

- Ukuran matriks `ubah` adalah (3, 2), yang berarti matriks 2 dimensi dengan 3 baris dan 2 kolom.

## Praktek 20:

```
# Praktek 20 : Vektor
# vektor baris
vek_1 = np.array([1, 2, 3])

# vektor kolom
vek_2 = np.array([[1],
                  [2],
                  [3]])
# atau menggunakan transpose()
vek_3 = np.array([[1, 2, 3]]).T

print("Vektor Baris")
print(vek_1)
print("vektor Kolom")
print(vek_2)
print("Vektor Kolom dengan transpose()")
print(vek_3)
```

Hasil:

```
Vektor Baris
[1 2 3]
vektor Kolom
[[1]
 [2]
 [3]]
Vektor Kolom dengan transpose()
[[1]
 [2]
 [3]]
```

Penjelasan:

1. `vek_1 = np.array([1, 2, 3])`:

- Baris ini membuat sebuah vektor baris `vek_1` dengan 3 elemen menggunakan numpy.
- Vektor baris adalah array 1 dimensi yang merepresentasikan vektor dalam ruang vektor.

**2. vek\_2 = np.array([[1], [2], [3]]):**

- Baris ini membuat sebuah vektor kolom vek\_2 dengan 3 elemen menggunakan numpy.
- Vektor kolom adalah array 2 dimensi yang merepresentasikan vektor dalam ruang vektor, dengan setiap elemen berada dalam baris yang berbeda.

**3. vek\_3 = np.array([[1, 2, 3]]).T:**

- Baris ini membuat sebuah vektor kolom vek\_3 dengan 3 elemen menggunakan numpy dan metode transpose().
- Metode transpose() mengubah vektor baris menjadi vektor kolom.

**4. print("Vektor Baris"):**

- Baris ini mencetak string "Vektor Baris".

**5. print(vek\_1):**

- Baris ini mencetak vektor baris vek\_1.

**6. print("vektor Kolom"):**

- Baris ini mencetak string "vektor Kolom".

**7. print(vek\_2):**

- Baris ini mencetak vektor kolom vek\_2.

**8. print("Vektor Kolom dengan transpose()"):**

- Baris ini mencetak string "Vektor Kolom dengan transpose()".

**9. print(vek\_3):**

- Baris ini mencetak vektor kolom vek\_3 yang dibuat menggunakan metode transpose().

## Praktek 21:

```
# Praktek 21 : Flatten()
# impor library numpy
import numpy as np

# membuat matriks
matriks_a = np.array([
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
])

# cetak matriks awal
print("Matriks Awal")
print(matriks_a)
print("Ukuran : ", matriks_a.shape)
print("\n")

# ubah matriks menjadi vektor
jd_vektor = matriks_a.flatten()

# cetak vektor
print("Hasil Konversi Matriks ke Vektor")
print(jd_vektor)
print("Ukuran : ", jd_vektor.shape)
```

### Hasil:

```
Matriks Awal
[[1 2 3]
 [4 5 6]
 [7 8 9]]
Ukuran :  (3, 3)

Hasil Konversi Matriks ke Vektor
[1 2 3 4 5 6 7 8 9]
Ukuran :  (9,)
```

### Penjelasan:

#### 1. import numpy as np:

- Baris ini mengimpor library numpy dan memberikan alias np untuk memudahkan penggunaan.

- Numpy adalah library Python yang digunakan untuk operasi numerik dan matriks.

2. `matriks_a = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])`:

- Baris ini membuat sebuah matriks 2 dimensi `matriks_a` dengan 3 baris dan 3 kolom menggunakan numpy.

- `np.array()` adalah fungsi yang digunakan untuk membuat array numpy.

3. `print("Matriks Awal")`:

- Baris ini mencetak string "Matriks Awal".

4. `print(matriks_a)`:

- Baris ini mencetak matriks `matriks_a`.

5. `print("Ukuran : ", matriks_a.shape)`:

- Baris ini mencetak ukuran matriks `matriks_a`, yang adalah (3, 3), yang berarti matriks 2 dimensi dengan 3 baris dan 3 kolom.

6. `jd_vektor = matriks_a.flatten()`:

- Baris ini mengubah matriks `matriks_a` menjadi vektor 1 dimensi menggunakan metode `flatten()`.

- Metode `flatten()` mengembalikan salinan array yang telah diratakan menjadi 1 dimensi.

7. `print("Hasil Konversi Matriks ke Vektor")`:

- Baris ini mencetak string "Hasil Konversi Matriks ke Vektor".

8. `print(jd_vektor)`:

- Baris ini mencetak vektor `jd_vektor` yang telah dihasilkan dari matriks `matriks_a`.

9. `print("Ukuran : ", jd_vektor.shape):`

- Baris ini mencetak ukuran vektor `jd_vektor`, yang adalah (9,), yang berarti vektor 1 dimensi dengan 9 elemen.

**SEKIAN DAN TERIMA KASIH**











