

**NAMA :IMAM ARDI PERDANA**

**NIM :24241020**

**KELAS :PTI A**

**MATKUL :STRUKTUR DATA**

## **MODUL 2**

### **Percobaan 1 : Array**

#### **Praktek 1**

```
Percobaan 1 array > array.py > ...
1  # Praktek 1 : Membuat array
2  # impor library numpy
3  import numpy as np
4
5  # membuat array dengan numpy
6  nilai_siswa = np.array([85, 55, 40, 90])
7
8  # akses data pada array
9  print(nilai_siswa[3])
```

#### **Hasil ouput**

```
PS E:\algoritma\Tugas-semester-2\Model 2> & C:/Users/USER/AppData/Local/Programs/Python/Python39-6/Scripts/python.exe E:\algoritma\Tugas-semester-2\Model 2/Percobaan 1 array/array.py
90
PS E:\algoritma\Tugas-semester-2\Model 2>
```

#### **Penjelasanya :**

##### **Baris 1:**

bahwa baris berikutnya akan mengimpor library NumPy, yang digunakan untuk operasi dan array.

##### **Baris 2:**

Mengimpor library NumPy dan memberi alias np supaya lebih ringkas saat digunakan. Setelah Bisa menggunakan np.array() untuk membuat array,bukan menulis numpy.array().

##### **Baris 3:**

menjelaskan bahwa baris di bawah akan membuat array menggunakan NumPy.

##### **Baris 4:**

Membuat sebuah array NumPy berisi nilai siswa: 85, 55, 40, dan 90. Array ini disimpan dalam Variable.

## Praktek 2

```
12 # Praktek 2 : Mengakses, Mengubah, dan Cek Ukuran dan Dimensi Array
13 # impor libaray numpy
14 import numpy as np
15
16 # membuat array dengan numpy
17 nilai_siswa_1 = np.array([75, 65, 45, 80])
18 nilai_siswa_2 = np.array([[85, 55, 40], [50, 40, 99]])
19
20 # cara akses elemen array
21 print(nilai_siswa_1[0])
22 print(nilai_siswa_2[1][1])
23
24 # mengubah nilai elemen array
25 nilai_siswa_1[0] = 88
26 nilai_siswa_2[1][1] = 70
27
28 # cek perubahannya
29 print(nilai_siswa_1[0])
30 print(nilai_siswa_2[1][1])
31
32 # Cek ukuran dan dimensi array
33 print("Ukuran Array : ", nilai_siswa_1.shape)
34 print("Ukuran Array : ", nilai_siswa_2.shape)
35 print("Dimensi Array : ", nilai_siswa_2.ndim)
```

## Hasil ouput

```
PS E:\algoritma\Tugas-semester-2\Model 2> & C:/Users/USER/AppData/Local/Model 2/Percobaan 1 array/aray.py"
90
75
40
88
70
Ukuran Array :  (4,)
Ukuran Array :  (2, 3)
Dimensi Array :  2
PS E:\algoritma\Tugas-semester-2\Model 2>
```

## Penjelasanya :

### Baris 1

yang menunjukkan bahwa baris selanjutnya akan mengimpor library NumPy

### Baris 2

Mengimpor library NumPy dan memberinya alias np agar lebih ringkas saat digunakan dalam Kode.

### Baris 3

bahwa kita akan membuat array menggunakan NumPy.

### Baris 4

Membuat array 1 dimensi dengan 4 elemen, lalu disimpan ke variabel nilai\_siswa\_1.

### Baris 5

Membuat array 2 dimensi (seperti matriks 2x3) yang disimpan dalam nilai\_siswa\_2.

## **Baris 6**

Yang menandai bahwa kita akan mengakses nilai-nilai dalam array.

## **Baris 7**

Menampilkan elemen pertama dari nilai\_siswa\_1, yaitu 75.

## **Baris 8**

Menampilkan baris ke-2, kolom ke-2 dari nilai\_siswa\_2, yaitu 40.

## **Baris 9**

bahwa kita akan mengubah isi array.

## **Baris 10**

Mengubah elemen pertama dari nilai\_siswa\_1 menjadi 88

## **Baris 11**

Mengubah elemen baris ke-2, kolom ke-2 dari nilai\_siswa\_2 menjadi 70.

## **Baris 12**

bahwa kita akan melihat apakah perubahan berhasil.

## **Baris 13**

Menampilkan elemen pertama dari nilai\_siswa\_1 yang sekarang sudah diubah menjadi 88.

## **Baris 14**

Menampilkan nilai pada nilai\_siswa\_2[1][1] yang sekarang menjadi 70.

## **Baris 15**

bahwa kita akan mengecek bentuk dan dimensi array.

## **Baris 16**

Menampilkan ukuran (jumlah elemen per dimensi) dari nilai\_siswa\_1. Hasil: (4,) → array 1 dimensi dengan 4 elemen.

## **Baris 17**

Menampilkan ukuran dari nilai\_siswa\_2.

Hasil: (2, 3) → array 2 dimensi, 2 baris dan 3 kolom.

## **Baris 18**

Menampilkan jumlah dimensi dari nilai\_siswa\_2, yaitu 2 (karena bentuknya seperti tabel/barisKolom).

## **Praktek 3**

```

38 # Praktek 3 : Operasi Aritmatika Pada Array
39 # impor library numpy
40 import numpy as np
41
42 # membuat array
43 a = np.array([1, 2, 3])
44 b = np.array([4, 5, 6])
45
46 # menggunakan operasi penjumlahan pada 2 array
47 print(a + b)      # array([5, 7, 9])
48
49 # Indexing dan Slicing pada Array
50 arr = np.array([10, 20, 30, 40])
51 print(arr[1:3])   # array([20, 30])
52
53
54 # iterasi pada array
55 for x in arr:
56     print(x)

```

Hasil outpunya

```

[5 7 9]
[20 30]
10
20
30
40
PS E:\algoritma\Tugas-semester-2\Model 2>

```

Penjelasannya :

Baris 1

Komentar yang menjelaskan bahwa library NumPy akan diimpor.

Baris 2

Mengimpor library NumPy dan memberi alias np supaya lebih singkat saat digunakan.

Baris 3

bahwa kita akan membuat array NumPy.

Baris 4

Membuat array a dengan elemen [1, 2, 3].

Baris 5

Membuat array b = np.array([4, 5, 6])

Baris 6

bahwa kita akan melakukan penjumlahan antar array

Baris 7

Menambahkan array a dan b secara elemen (element-wise):  $[1+4, 2+5, 3+6] \rightarrow [5, 7, 9]$ .

Baris 8

bahwa baris berikut akan menunjukkan teknik mengambil sebagian isi array.

Baris 9

Membuat array arr dengan 4 elemen: [10, 20, 30, 40]

Baris 10

Mengambil elemen dari indeks ke-1 hingga sebelum ke-3 (slicing): arr[1:3] → [20, 30].

Baris 11

bahwa kita akan melakukan iterasi (perulangan) pada elemen array

Baris 12–13

```
for x in arr: print(x)
```

Melakukan loop untuk mencetak setiap elemen dalam array arr.

## Percobaan 2 : Traversal, Insertion, Deletion

### Praktek 4

```
Percobaan 2 Traversal, Insertion, Deletion > Traversal, Insertion, Deletion.py > ...
1  # Praktek 4 : Linear Traversal
2  # membuat array
3  arr = [1, 2, 3, 4, 5]
4
5  # Linear Traversal ke tiap elemen arr
6  print("Linear Traversal: ", end=" ")
7  for i in arr:
8      print(i, end=" ")
9  print()
```

Hasil outpunya

```
PS E:\algoritma\Tugas-semester-2\Model 2> & C:/Users/US
Model 2/Percobaan 2 Traversal, Insertion, Deletion/Trav
Linear Traversal:  1 2 3 4 5
PS E:\algoritma\Tugas-semester-2\Model 2> █
```

Penjelsanyan :

Baris 1

Komentar yang menjelaskan bahwa kamu akan membuat array (dalam bentuk list di Python, Bukan numpyarray).

Baris 2

Membuat list bernama arr yang berisi lima elemen: [1, 2, 3, 4, 5].

Baris 3

Komentar bahwa kamu akan melakukan traversal linear, yaitu mengunjungi dan memproses Elemen satu persatu dari ke kiri ke kanan.

Baris 4

Mencetak teks "Linear Traversal: " tanpa pindah ke baris baru (karena end=" " membuat kursor Tetap dibaris yang sama dan menambahkan spasi).

Baris 5–6

```
for i in arr: print(i, end="
```

```
print(i, end=" ")
```

 mencetak setiap elemen diikuti oleh spasi, bukan pindah baris.

Baris 7

Mencetak baris kosong untuk pindah ke baris baru setelah selesai Mencetak semua elemen baru.

Linear Traversal: 1 2 3 4 5

Teks "Linear Traversal: " dicetak terlebih dahulu.

Kemudian setiap elemen 1 2 3 4 5 dicetak di baris yang sama, dipisahkan oleh spasi. Setelah selesai, baris kosong ditambahkan dengan print() untuk menjaga format tampilan.

## Praktek 5

```
11 # Praktek 5 : Reverse Traversal
12 # membuat array
13 arr = [1, 2, 3, 4, 5]
14
15 # Reverse Traversal dari elemen akhir
16 print("Reverse Traversal: ", end="")
17 for i in range(len(arr) - 1, -1, -1):
18     print(arr[i], end=" ")
19 print()
```

Hasil outpunya

```
Reverse Traversal: 5 4 3 2 1
PS E:\algoritma\Tugas-semester-2\Model 2>
```

Penjelasnya :

Baris 1

Komentar bahwa kamu akan membuat array (list) di baris berikutnya.

Baris 2

Membuat list arr yang berisi lima elemen dari 1 sampai 5.

Baris 3

Komentar bahwa kamu akan mencetak elemen dari list arr secara terbalik (dari belakang Ke depan).

Baris 4

Mencetak teks "Reverse Traversal: " tanpa pindah baris karena end="" menjaga agar output Selanjutnya dicetak dibaris yang sama.

Baris 5

$\text{len}(\text{arr}) - 1 = 4 \rightarrow$  indeks terakhir (karena jumlah elemen 5 dan indeks mulai dari 0).

-1 adalah batas akhir (exclusive)  $\rightarrow$  berarti iterasi akan berhenti sebelum mencapai indeks -1, Alias berhenti di 0

-1 adalah langkah (step)  $\rightarrow$  artinya mundur satu per satu. Jadi,  $\text{range}(4, -1, -1)$  menghasilkan urutan indeks: 4, 3, 2, 1, 0

Baris 6

Untuk setiap indeks i, ambil elemen  $\text{arr}[i]$  lalu cetak di baris yang sama, dipisahkan dengan spasi.

Baris 7

Pindah ke baris baru setelah mencetak semua elemen, agar output rapi.

Praktek 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 ouputnya

```
Linear Traversal using while loop: 1 2 3 4 5
PS E:\algoritma\Tugas-semester-2\Model 2>
```

Penjelasanya :

Baris 1

bahwa kamu akan membuat array (list).

Baris 2

Membuat list arr berisi 5 elemen: [1, 2, 3, 4, 5].

Baris 3

bahwa variabel-variabel awal akan didefinisikan.

Baris 4

n menyimpan panjang (jumlah elemen) dari array arr, yaitu 5.

Baris 5

Variabel i digunakan sebagai indeks awal untuk perulangan. Dimulai dari 0 (indeks pertama Array).

Baris 6

Mencetak teks pembuka, tanpa pindah baris, karena end=" " menjaga agar output berikutnya Tetap di baris yang sama

Baris 7

akan menggunakan perulangan while untuk traversal.

Baris 8–10

Perulangan akan berjalan selama i kurang dari n (panjang array). arr[i] mencetak elemen ke-i dari array.

end=" " agar semua elemen dicetak dalam satu baris, dipisahkan spasi. i += 1 menaikkan indeks agar pindah ke elemen berikutnya.

Loop ini mencetak 1 2 3 4 5

Baris 11

Pindah ke baris baru setelah selesai mencetak elemen array.

## Praktek 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 ouputnya

```
Reverse Traversal using while loop: [5, 4, 3, 2, 1]
PS E:\algoritma\Tugas-semester-2\Model 2>
```

Penjelasanya :

Baris 1

membuat sebuah array (list).

Baris 2



Membuat list bernama arr berisi elemen [1, 2, 3, 4, 5].

Baris 3

menetapkan variabel awal untuk indeks traversal.

Baris 4–5

start diset ke indeks pertama (0).

end diset ke indeks terakhir ( $\text{len}(\text{arr}) - 1 = 4$ ).

Variabel ini akan digunakan untuk menukar elemen dari ujung ke tengah.

Baris 6

Mencetak teks sebagai keterangan, tanpa pindah ke baris baru (`end=" "`).

Baris 7

melakukan pembalikan isi array dengan perulangan while.

Baris 8–11

Loop akan terus berjalan selama  $\text{start} < \text{end}$ . Di dalam loop:

Elemen pada posisi start dan end ditukar (swap).

Kemudian start maju ke kanan (+1) dan end mundur ke kiri (-1).

Proses ini membalik urutan elemen dari luar ke dalam.

1.  $\text{start}=0, \text{end}=4$ : tukar 1 dan 5  $\rightarrow [5, 2, 3, 4, 1]$

2.  $\text{start}=1, \text{end}=3$ : tukar 2 dan 4  $\rightarrow [5, 4, 3, 2, 1]$

3.  $\text{start}=2, \text{end}=2$ : kondisi  $\text{start} < \text{end}$  sudah tidak terpenuhi, loop berhenti.

Baris 12

Mencetak isi array setelah dibalik. Hasil akhirnya:

[5, 4, 3, 2, 1]

## Praktek 8

```
# 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 ouputnya

```
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  
PS E:\algoritma\Tugas-semester-2\Model 2>
```

### Penjelasannya :

#### Baris 1

membuat array (dalam Python disebut list).

#### Baris 2

Membuat list arr dengan 6 elemen angka: [12, 16, 20, 40, 50, 70]

#### Baris 3

mencetak isi array sebelum elemen baru disisipkan.

#### Baris 4

Mencetak isi list arr sebelum ada perubahan:

Output: Array Sebelum Insertion : [12, 16, 20, 40, 50, 70]

#### Baris 5

mencetak jumlah elemen list sebelum penambahan.

#### Baris 6

Menggunakan len(arr) untuk menghitung jumlah elemen, yaitu 6.

Output: Panjang Array : 6

#### Baris 7

menambahkan elemen di akhir list dengan fungsi .append().

#### Baris 8

Menambahkan angka 26 ke akhir list arr.

List berubah menjadi: [12, 16, 20, 40, 50, 70, 26]

#### Baris 9

mencetak array setelah penambahan elemen.

#### Baris 10

Mencetak isi array setelah penambahan output: array setelah insertion : [12, 16, 20, 40, 50, 70, 26]

#### Baris 11

mencetak jumlah elemen setelah penambahan.

#### Baris 12

Mencetak jumlah elemen saat ini, yaitu 7.

**Output: Panjang Array : 7**

## **Praktek 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 ouputnya**

```
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
PS E:\algoritma\Tugas-semester-2\Model 2>
```

## **Penjelasannya :**

### **Baris 1**

membuat array (dalam Python disebut list).

### **Baris 2**

Membuat list arr dengan 6 elemen angka: [12, 16, 20, 40, 50, 70]

### **Baris 3**

mencetak isi array sebelum elemen baru disisipkan.

### **Baris 4**

Mencetak isi list arr sebelum ada perubahan:

Output:

Array Sebelum Insertion : [12, 16, 20, 40, 50, 70]

### **Baris 5**

mencetak jumlah elemen list sebelum penambahan.

### **Baris 6**

Menggunakan len(arr) untuk menghitung jumlah elemen, yaitu 6.

Output:

Panjang Array : 6

Baris 7

menambahkan elemen di akhir list dengan fungsi .append().

Baris 8

Menambahkan angka 26 ke akhir list arr.

List berubah menjadi: [12, 16, 20, 40, 50, 70, 26]

Baris 9

mencetak array setelah penambahan elemen.

Baris 10

Mencetak isi array setelah penambahan output: array setelah insertion : [12, 16, 20, 40, 50, 70, 26]

Baris 11

mencetak jumlah elemen setelah penambahan.

Baris 12

Mencetak jumlah elemen saat ini, yaitu 7.

Output:

Panjang Array : 7

## Praktek lanjutan 10

```
# 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 ouputnya

```
Array Sebelum Penyisipan : [12, 16, 20, 40, 50, 70]  
Array Sesudah Penyisipan : [12, 16, 20, 40, 5, 50, 70]  
PS E:\algoritma\Tugas-semester-2\Modul 2>
```

### **Penjelasannya :**

#### **Baris 1**

**Komentar bahwa kode tidak menggunakan fungsi .insert() bawaan Python.**

#### **Baris 2**

**Komentar bahwa kode akan membuat array dan mencetaknya.**

#### **Baris 3**

**Membuat array/list bernama arr yang berisi angka: 12, 16, 20, 40, 50, 70.**

#### **Baris 4**

**Mencetak isi array sebelum penyisipan dilakukan.**

#### **Baris 5**

**Komentar bahwa akan ditentukan posisi penyisipan elemen baru di tengah array.**

#### **Baris 6**

**Menetapkan nilai pos = 4, artinya elemen baru akan disisipkan di indeks ke-4.**

#### **Baris 7**

**Komentar bahwa akan ditentukan nilai yang akan disisipkan.**

#### **Baris 8**

**Menetapkan nilai x = 5, yaitu nilai yang akan disisipkan ke dalam array.**

#### **Baris 9**

**Komentar bahwa panjang array akan ditambah dengan elemen dummy agar bisa digeser.**

#### **Baris 10**

**Menambahkan elemen dummy 0 ke akhir array untuk memberi ruang penyisipan.**

#### **Baris 11**

**Komentar bahwa elemen-elemen akan digeser dari belakang ke depan.**

#### **Baris 12**

**Melakukan perulangan dari indeks sebelum elemen dummy sampai ke posisi sisip, secara mundur.**

#### **Baris 13**

**Menggeser elemen satu posisi ke kanan untuk memberi ruang di posisi pos.**

#### **Baris 14**

**Komentar bahwa nilai baru akan dimasukkan ke posisi yang telah disediakan.**

#### **Baris 15**

**Menyisipkan nilai x (yaitu 5) ke dalam array pada indeks ke-4.**

#### **Baris 16**

**Komentar bahwa array baru akan dicetak setelah penyisipan.**

## Baris 17

Mencetak isi array setelah elemen baru berhasil disisipkan.

## Praktek 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 ouputnya

```
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]
PS E:\algoritma\Tugas-semester-2\Modul 2>
```

## Penjelasannya :

### Baris 1

Komentar judul praktek: "Praktek 11: Menghapus array".

### Baris 2

Komentar bahwa akan membuat array terlebih dahulu.

### Baris 3

Membuat array a berisi lima elemen: 10, 20, 30, 40, dan 50.

### Baris 4

Mencetak array sebelum dilakukan penghapusan elemen.

### Baris 5

Komentar bahwa akan menghapus elemen pertama yang bernilai 30 (berdasarkan nilai, bukan indeks).

### Baris 6

Menghapus elemen bernilai 30 menggunakan method .remove(30).

### Baris 7

Mencetak array setelah elemen 30 dihapus.

#### Baris 8

Komentar bahwa akan menghapus elemen berdasarkan indeks, yaitu elemen pada indeks ke-1 (yaitu 20 setelah 30 dihapus).

#### Baris 9

Menghapus dan menyimpan elemen di indeks ke-1 menggunakan `.pop(1)`.

#### Baris 10

Mencetak elemen yang telah di-*pop* (dihapus dan disimpan ke variabel).

#### Baris 11

Mencetak array setelah penghapusan elemen dengan `.pop()`.

#### Baris 12

Komentar bahwa akan menghapus elemen pertama (yang sekarang bernilai 10).

#### Baris 13

Menghapus elemen pertama menggunakan perintah `del`.

#### Baris 14

Mencetak array setelah elemen pertama dihapus menggunakan `del`.

### Praktek 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 ouputnya

```
PS E:\algoritma\Tugas-semester-2\Modul 2> & C:/Users/USER/AppData/Local/Programs/Python/Python
Modul 2/Percobaan 3 Matriks atau Array 2 dan 3 Dimensi/Matriks atau Array 2 dan 3 Dimensi.py"
9
PS E:\algoritma\Tugas-semester-2\Modul 2>
```

#### Penjelasannya :

##### Baris 1

Komentar judul praktek: *Praktek 12: Membuat Matriks 2 Dimensi*.

##### Baris 2

Komentar bahwa program akan mengimpor library NumPy.

##### Baris 3

Mengimpor library NumPy dan memberi alias np agar lebih ringkas saat digunakan.

#### Baris 4

Komentar bahwa akan membuat matriks menggunakan fungsi dari NumPy.

#### Baris 5

Membuat array 2 dimensi (matriks) dengan 3 baris dan 3 kolom:

- Baris 1: [1, 2, 3]
- Baris 2: [4, 5, 6]
- Baris 3: [7, 8, 9]

Array ini disimpan dalam variabel `matriks_np`.

#### Baris 6

Mengakses dan mencetak elemen pada baris ke-3 dan kolom ke-3 dari matriks, yaitu 9.

(Penulisan indeks dimulai dari 0, sehingga `matriks_np[2][2]` menunjuk ke elemen baris ke-3 dan kolom ke-3.)

#### Praktek 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 ouputnya

```
Hasil Penjumlahan Matriks dari LIST
[17, 15, 4]
[10, 12, 9]
[11, 13, 18]
PS E:\algoritma\Tugas-semester-2\Modul 2>
```

Penjelasannya :



**Baris 1**

**Komentar judul praktek: *Praktek 13: Operasi Penjumlahan Matriks dengan list.***

**Baris 2**

**Komentar bahwa program akan melakukan penjumlahan dua buah matriks yang didefinisikan menggunakan struktur data list.**

**Baris 3 - 6**

**Membuat matriks X yang terdiri dari 3 baris dan 3 kolom:**

- **Baris 1: [12, 7, 3]**
- **Baris 2: [4, 5, 6]**
- **Baris 3: [7, 8, 9]**

**Baris 7 - 10**

**Membuat matriks Y yang juga terdiri dari 3 baris dan 3 kolom:**

- **Baris 1: [5, 8, 1]**
- **Baris 2: [6, 7, 3]**
- **Baris 3: [4, 5, 9]**

**Baris 11 - 14**

**Membuat matriks result dengan ukuran yang sama seperti X dan Y, berisi nilai awal 0 sebagai tempat untuk menyimpan hasil penjumlahan.**

**Baris 15**

**Komentar bahwa akan dilakukan proses penjumlahan dua matriks menggunakan perulangan bertingkat (nested loop).**

**Baris 16**

**Melakukan perulangan untuk setiap baris (row) dalam matriks menggunakan range(len(X)).**

**Baris 17**

**Melakukan perulangan untuk setiap kolom dalam satu baris menggunakan range(len(X[0])).**

**Baris 18**

**Melakukan penjumlahan elemen dari matriks X dan Y pada indeks [i][j] lalu menyimpannya ke result[i][j].**

**Baris 20**

**Mencetak judul dari hasil penjumlahan matriks.**

**Baris 23**

**Melakukan pencetakan isi dari matriks result satu baris per iterasi.**

**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 ouputnya

```
Hasil Penjumlahan Matriks dari NumPy
[[17 15  4]
 [10 12  9]
 [11 13 18]]
PS E:\algoritma\Tugas-semester-2\Modul 2>
```

Penjelasannya :

Baris 1

Komentar judul praktek: *Praktek 14: Operasi Penjumlahan Matriks dengan NumPy*.

Baris 2

Komentar bahwa akan mengimpor library numpy.

Baris 3

Mengimpor library numpy dengan alias np.

Membuat Matriks dengan NumPy

Baris 5

Komentar bahwa akan membuat matriks menggunakan NumPy.

Baris 6 - 9

Membuat matriks X yang berisi:

- Baris 1: [12, 7, 3]
- Baris 2: [4, 5, 6]
- Baris 3: [7, 8, 9]

Baris 10 - 13

Membuat matriks Y yang berisi:

- Baris 1: [5, 8, 1]
- Baris 2: [6, 7, 3]

- Baris 3: [4, 5, 9]

## Operasi Penjumlahan

### Baris 15

Komentar bahwa akan dilakukan penjumlahan dua buah matriks.

### Baris 16

Melakukan penjumlahan langsung antara X dan Y menggunakan operator +. Karena menggunakan NumPy, operasi ini dilakukan secara elemen-wise (elemen per elemen).

## Menampilkan Hasil

### Baris 18

Komentar bahwa hasil akan dicetak ke layar.

### Baris 19

Mencetak judul hasil penjumlahan matriks.

### Baris 20

Mencetak isi dari matriks hasil (result).

## Praktek 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 ouputnya

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

Penjelasannya :

**Baris 1**

**Komentar judul praktek: *Praktek 15: Operasi Pengurangan Matriks dengan NumPy*.**

**Baris 2**

**Komentar bahwa akan mengimpor library numpy.**

**Baris 3**

**Mengimpor library numpy dengan nama alias np, yang umum digunakan.**

**Membuat Matriks dengan NumPy**

**Baris 5**

**Komentar bahwa akan dibuat dua buah matriks.**

**Baris 6 - 9**

**Membuat matriks X dengan isi:**

- **Baris 1: [12, 7, 3]**
- **Baris 2: [4, 5, 6]**
- **Baris 3: [7, 8, 9]**

**Baris 10 - 13**

**Membuat matriks Y dengan isi:**

- **Baris 1: [5, 8, 1]**
- **Baris 2: [6, 7, 3]**
- **Baris 3: [4, 5, 9]**

**Operasi Pengurangan**

**Baris 15**

**Komentar bahwa akan dilakukan pengurangan dua buah matriks.**

**Baris 16**

**Melakukan operasi pengurangan elemen per elemen:  $\text{result} = X - Y$ .**

**Menampilkan Hasil**

**Baris 18**

**Komentar bahwa hasil akan ditampilkan di layar.**

**Baris 19**

**Mencetak teks judul hasil pengurangan.**

**Baris 20**

**Mencetak matriks hasil pengurangan (result).**

**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 list")
print(result)
```

Hasil ouputnya

```
Hasil Perkalian Matriks dari list
[[60 56  3]
 [24 35 18]
 [28 40 81]]
PS E:\algoritma\Tugas-semester-2\Modul 2>
```

Penjelasannya :

Baris 1

Komentar judul: *Praktek 16: Operasi Perkalian Matriks dengan NumPy.*

Baris 2

Komentar bahwa akan digunakan pustaka numpy.

Baris 3

Mengimpor library numpy dan memberi alias np (singkatan umum).

Membuat Matriks

Baris 5

Komentar bahwa akan dibuat dua matriks.

Baris 6–9

Membuat matriks X:

[[12, 7, 3],

[ 4, 5, 6],

[ 7, 8, 9]]

Baris 10–13

Membuat matriks Y:

[[5, 8, 1],

[6, 7, 3],

[4, 5, 9]]

## Operasi Perkalian Elemen-wise (Hadamard Product)

Baris 15

Komentar bahwa akan dilakukan operasi perkalian.

Baris 16

```
result = X * Y
```

Melakukan perkalian elemen-wise, artinya setiap elemen pada posisi yang sama di kedua matriks dikalikan:

```
result[0][0] = 12 * 5 = 60
```

```
result[0][1] = 7 * 8 = 56
```

```
result[0][2] = 3 * 1 = 3
```

...

Baris 18

Komentar untuk menampilkan hasil.

Baris 19

```
print("Hasil Perkalian Matriks dari list")
```

Teks ini sebenarnya kurang tepat karena operasi ini menggunakan NumPy, bukan list biasa. Sebaiknya ditulis:

```
print("Hasil Perkalian Matriks dari NumPy")
```

Baris 20

```
print(result)
```

Menampilkan hasil perkalian elemen-wise.

## Praktek 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 ouputnya

```
Hasil Pembagian Matriks dari NumPy
[[2.4      0.875    3.      ]
 [0.66666667 0.71428571 2.      ]
 [1.75     1.6      1.      ]]
PS E:\algoritma\Tugas-semester-2\Modul 2>
```

**Penjelasannya :**

**Baris 1:**

Program ini diberi judul "Praktek 17: Operasi Pembagian Matriks dengan NumPy", yang menunjukkan bahwa praktek ini bertujuan mempelajari cara melakukan pembagian antar elemen matriks menggunakan library NumPy.

**Baris 2:**

Baris ini hanya komentar yang menjelaskan bahwa di bawah ini akan dilakukan proses impor pustaka NumPy.

**Baris 3:**

Pustaka NumPy diimpor dan diberi alias np. NumPy adalah library Python yang sangat umum digunakan untuk operasi matematika, terutama pengolahan array dan matriks.

**Baris 4:**

Komentar ini memberi tahu bahwa setelah ini program akan membuat matriks menggunakan NumPy.

**Baris 5–9:**

Pada bagian ini dibuat sebuah matriks bernama X yang berukuran 3 baris dan 3 kolom. Matriks ini berisi nilai-nilai angka sebagai data awal yang nantinya akan dibagi.

**Baris 10–14:**

Dibuat lagi sebuah matriks lain bernama Y dengan ukuran yang sama yaitu 3x3. Matriks ini berfungsi sebagai penyebut dalam operasi pembagian antar elemen.

**Baris 15:**

Komentar yang menjelaskan bahwa di bawah ini akan dilakukan operasi pembagian antar dua matriks.

**Baris 16:**

Program melakukan pembagian antar elemen yang sesuai posisi antara matriks X dan Y. Misalnya, elemen baris pertama kolom pertama di X dibagi dengan elemen baris pertama kolom pertama di Y, dan seterusnya. Hasil dari pembagian disimpan dalam variabel baru bernama result.

**Baris 17:**

Komentar bahwa bagian berikutnya akan mencetak hasil ke layar.

**Baris 18:**

Program mencetak teks "Hasil Pembagian Matriks dari NumPy" ke layar sebagai keterangan dari hasil yang ditampilkan.

**Baris 19:**

Program mencetak isi dari variabel result, yaitu hasil pembagian dari setiap elemen 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 ouputnya

```
Matriks Sebelum Transpose
[[1 2 3]
 [4 5 6]
 [7 8 9]]
Matriks Setelah Transpose
[[1 4 7]
 [2 5 8]
 [3 6 9]]
PS E:\algoritma\Tugas-semester-2\Modul 2>
```

### Penjelasannya :

#### Baris 1:

Program diberi judul “Praktek 18: transpose()”, yang menunjukkan bahwa tujuan utama adalah mempelajari bagaimana melakukan transpose pada matriks.

#### Baris 2:

Komentar bahwa program akan mengimpor pustaka NumPy.

#### Baris 3:

Library NumPy diimpor dan diberi alias np agar pemanggilan fungsinya lebih ringkas dan mudah digunakan.

#### Baris 4:

Komentar bahwa baris berikutnya akan membuat sebuah matriks.

#### Baris 5–9:

Dibuat sebuah matriks bernama matriks\_a dengan ukuran 3 baris dan 3 kolom, yang berisi angka-angka dari 1 hingga 9. Matriks ini adalah data awal yang akan di-transpose.

#### Baris 10:

Komentar bahwa bagian ini akan mencetak matriks sebelum dilakukan transpose.



**Baris 11–12:**

Program menampilkan teks “Matriks Sebelum Transpose”, kemudian mencetak isi dari matriks\_a ke layar agar terlihat bentuk awal matriks.

**Baris 13:**

Komentar yang menjelaskan bahwa bagian ini akan melakukan proses transpose terhadap matriks.

**Baris 14:**

Program melakukan transpose terhadap matriks\_a, yaitu menukar baris menjadi kolom dan kolom menjadi baris. Hasilnya disimpan ke variabel baru bernama balik.

**Baris 15:**

Komentar bahwa bagian ini akan mencetak matriks setelah dilakukan proses transpose.

**Baris 16–17:**

Program mencetak teks “Matriks Setelah Transpose”, lalu menampilkan isi dari variabel balik, yaitu hasil dari transpose matriks.

## 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 ouputnya

```
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)
```

```
PS E:\algoritma\Tugas-semester-2\Modul 2> █
```

**Penjelasannya :**

**Baris 1:**

Judul program ditulis sebagai “Praktek 19: reshape()”, yang menunjukkan bahwa program ini membahas tentang cara mengubah bentuk (dimensi) array menggunakan fungsi reshape() dari NumPy.

**Baris 2:**

Komentar bahwa bagian berikutnya akan mengimpor library NumPy.

**Baris 3:**

Library NumPy diimpor dengan alias np agar lebih praktis saat digunakan dalam kode.

**Baris 4:**

Komentar bahwa di bawah ini akan dibuat sebuah array satu dimensi.

**Baris 5:**

Dibuat array satu dimensi bernama arr\_1d yang berisi enam elemen numerik. Ini adalah array awal sebelum diubah bentuknya (reshape).

**Baris 6:**

Komentar bahwa bagian ini akan mencetak isi array sebelum diubah bentuknya.

**Baris 7–9:**

Program mencetak teks “Matriks Sebelum Reshape”, menampilkan isi array arr\_1d, dan menampilkan ukuran array tersebut dengan properti .shape. Ukuran awal array adalah (6,) karena satu dimensi dengan enam elemen.

**Baris 10:**

Mencetak baris kosong (newline) agar hasil di layar lebih rapi dan mudah dibaca.

**Baris 11:**

Komentar bahwa bagian ini akan mengubah bentuk (reshape) array dari 1 dimensi menjadi array 2 dimensi berukuran 3 baris dan 2 kolom.

**Baris 12:**

Fungsi .reshape(3, 2) digunakan untuk mengubah array arr\_1d menjadi array dua dimensi dengan ukuran 3x2. Hasilnya disimpan ke variabel ubah.

**Baris 13:**

Komentar bahwa bagian ini akan mencetak hasil reshape ke layar.

**Baris 14–15:**

Program mencetak teks “Matriks Setelah Reshape”, menampilkan isi array setelah diubah bentuknya, dan mencetak ukurannya menggunakan .shape, yang kini akan menjadi (3, 2).

**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 ouputnya

```

Vektor Baris
[1 2 3]
vektor Kolom
[[1]
 [2]
 [3]]
Vektor Kolom dengan transpose()
[[1]
 [2]
 [3]]
PS E:\algoritma\Tugas-semester-2\Modul 2>

```

## Penjelasannya :

### Baris 1:

Judul program ditulis sebagai “Praktek 20: Vektor” yang menunjukkan bahwa praktik ini berfokus pada pembuatan dan manipulasi vektor dalam NumPy.

### Baris 2:

Komentar bahwa bagian berikutnya akan membuat vektor baris.

### Baris 3:

vek\_1 adalah vektor baris, dibuat menggunakan np.array([1, 2, 3]). Artinya, array ini terdiri dari satu baris dan tiga kolom (bentuk 1x3).

### Baris 5:

Komentar bahwa bagian ini akan membuat vektor kolom.

### Baris 6–8:

vek\_2 adalah vektor kolom, dibuat sebagai array dua dimensi dengan masing-masing elemen diletakkan dalam satu baris terpisah, sehingga membentuk kolom. Bentuk array-nya adalah 3x1.

### Baris 9:

Komentar alternatif untuk membuat vektor kolom menggunakan metode transpose.

### Baris 10:

vek\_3 dibuat dari array [1, 2, 3] yang ditranspose (.T), sehingga dari bentuk baris (1x3) diubah menjadi kolom (3x1).

**Baris 12:**

**Mencetak judul: “Vektor Baris”.**

**Baris 13:**

**Mencetak isi dari vek\_1, yaitu vektor baris.**

**Baris 14:**

**Mencetak judul: “Vektor Kolom”.**

**Baris 15:**

**Mencetak isi dari vek\_2, yaitu vektor kolom yang dibuat langsung sebagai array 2 dimensi.**

**Baris 16:**

**Mencetak judul: “Vektor Kolom dengan transpose()”.**

**Baris 17:**

**Mencetak isi dari vek\_3, yaitu vektor kolom yang dibentuk dari vektor baris dengan cara 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 ouputnya

```
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,)
PS E:\algoritma\Tugas-semester-2\Modul 2>
```

**Penjelasannya :**

**Baris 1:**

Judul praktik ditulis sebagai “Praktek 21: Flatten()”, yang menjelaskan bahwa fokusnya adalah penggunaan fungsi `.flatten()` pada array NumPy.

**Baris 2–3:**

Mengimpor library numpy dengan alias `np` untuk digunakan dalam pengolahan array.

**Baris 5:**

Komentar bahwa bagian ini digunakan untuk membuat matriks.

**Baris 6–10:**

Membuat array dua dimensi bernama `matriks_a` dengan bentuk 3 baris dan 3 kolom:

`[[1, 2, 3],`

`[4, 5, 6],`

`[7, 8, 9]]`

**Baris 12:**

Mencetak judul “Matriks Awal”.

**Baris 13:**

Menampilkan isi dari `matriks_a` (sebelum diubah).

**Baris 14:**

Menampilkan ukuran (`shape`) dari matriks menggunakan `matriks_a.shape`, yang akan menghasilkan (3, 3).

**Baris 15:**

Memberikan baris kosong agar hasil cetakan lebih rapi.

**Baris 17:**

Membuat array baru `jd_vektor` dari `matriks_a` dengan metode `.flatten()`, yaitu mengubah array 2 dimensi menjadi array 1 dimensi (vektor).

Contoh hasil: `[1 2 3 4 5 6 7 8 9]`

**Baris 19:**

Mencetak judul “Hasil Konversi Matriks ke Vektor”.

**Baris 20:**

Menampilkan isi dari `jd_vektor`, yaitu hasil dari proses flattening.

**Baris 21:**

Menampilkan ukuran `jd_vektor` dengan `.shape`, yang akan menjadi (9,) karena array-nya sekarang 1 dimensi dengan 9 elemen.