

NAMA : BAIQ AURA
RACHMI WIJAYANTHI
NIM : 24241014

MATKUL : STRUKTUR DATA

Praktek 1

main.py

+

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

Outputnya :

90

** Process exited - Return Code: 0 **

Press Enter to exit terminal

Penjelasannya :

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.

Penjelasan outputnya :

Program mencetak 90 ke layar karena itu adalah nilai pada indeks ke-3 dari array cara perhitunganya dari nol mulai dari indeks tersebut maka hasil akhirnya 90.

Praktek 2

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

# cek perubahannya dengan akses elemen array
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)
```

Outputnya :

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

Penjelasannya :

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/baris-Kolom).

Praktek 3

```
# 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])
```

Outputnya :

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

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]` \rightarrow [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.

Praktek 4**Metode 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()
```

Outputnya :

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

```
** Process exited - Return Code: 0 **
```

```
Press Enter to exit terminal
```

Penjelasannya :**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

```
# 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()
```

Outputnya :

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

```
** Process exited - Return Code: 0 **
```

```
Press Enter to exit terminal
```

Penjelasannya :

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

$\text{len(arr)} - 1 = 4 \rightarrow$ 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.

Penjelasan outputnya :

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

Program mencetak elemen dari indeks terakhir ($\text{arr}[4] = 5$) sampai indeks pertama ($\text{arr}[0] = 1$)

Secara mundur

Semuanya dicetak dalam satu baris setelah teks "Reverse Traversal: ".

Praktek 7

```
# 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()
```

Outputnya :

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

```
** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

Penjelasannya :

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.

Penjelasan outputnya :

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

Program menelusuri list dari elemen pertama hingga terakhir menggunakan while, dan Mencetak semua elemen secara berurutan.

Praktek 8

```
# 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:

    arr[start], arr[end] = arr[end], arr[start]
    start += 1
    end -= 1
print(arr)
```

Outputnya :

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

```
** Process exited - Return Code: 0 **
```

```
Press Enter to exit terminal
```

Penjelasannya :

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 (len(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 start < 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. start=0, end=4: tukar 1 dan 5 → [5, 2, 3, 4, 1]

2. start=1, end=3: tukar 2 dan 4 → [5, 4, 3, 2, 1]

3. start=2, end=2: kondisi start < end sudah tidak terpenuhi, loop berhenti.

Baris 12

Mencetak isi array setelah dibalik. Hasil akhirnya:

```
[5, 4, 3, 2, 1]
```

Penjelasan outputnya :

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

Array awal [1, 2, 3, 4, 5] dibalik urutannya menjadi [5, 4, 3, 2, 1].

Proses ini disebut reverse in-place karena dilakukan langsung di array yang sama tanpa Membuat array baru.

Praktek 9

```
# 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))
```

Outputnya :

```
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
```

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

Penjelasan outputnya :

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

Praktek 10

```
# 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))
```

Outputnya :

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

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

Penjelasan outputnya :

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