

TUGAS PRAKTEK 1-10 STRUKTUR DATA



MATA KULIAH STRUKTUR DATA

Ditulis oleh :

Ezi andrean (24241027)

PRAKTEK 1:

```
# 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 :

impor library numpy

- **Artinnya** : untuk mengubah atau mempersingkat kata numpy menjadi np.

membuat array dengan numpy

- **Artinya** : untuk menginput code np saat pembuatan array.

akses pada array

- **Artinya** : seperti `print(nilai_siswa[3])`, karna disini disuruh mengakses data array di list ke 3, karna list di mulai dari 0 maka diatas itu list 85 adalah 0 dan 55 adalah 1 dan seterusnya, maka jika disuruh print nilai dari list ketiga adalah 90.

PRAKTEK2:

```
# 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 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)
```

Hasil:

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

Penjelasan :

impor library numpy

- **Artinya:** Mengimpor *library* bernama **NumPy**, yang digunakan untuk mengolah array atau data numerik.
- `numpy as np` berarti kamu bisa menyebut numpy cukup dengan `np` agar lebih singkat.

membuat array dengan numpy

- **Artinya:** Membuat struktur data **array** (seperti daftar angka) menggunakan NumPy.
- `np.array()` adalah fungsi untuk membuat array.

cara akses elemen array

- **Artinya:** Menunjukkan cara **mengambil nilai tertentu** dari dalam array.
- Misalnya: `nilai_siswa_1[0]` berarti ambil elemen pertama dari array `nilai_siswa_1`.

mengubah nilai elemen array

- **Artinya:** Menunjukkan cara **mengganti nilai** dari elemen array tertentu.
- Misalnya: `nilai_siswa_1[0] = 88` berarti nilai pertama diganti jadi 88.

cek perubahannya dengan akses elemen array

- **Artinya:** Mengecek apakah perubahan nilai yang sudah dilakukan berhasil.
- Dengan mencetak kembali elemen yang diubah.

Cek ukuran dan dimensi array

- **Artinya:** Melihat **ukuran array** (berapa banyak elemen/baris/kolom) dan **berapa dimensi** array-nya.
- `.shape` → memberi informasi bentuk array (misalnya 1 baris 4 kolom).
- `.ndim` → memberi informasi jumlah dimensi array (misalnya 1D, 2D, dst).

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])

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

Hasil :

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

Penjelasan :

impor library numpy

- Artinya: Kita mengambil **library numpy**, yaitu alat bantu khusus untuk mengolah angka dan array (seperti daftar angka).
- as np artinya kita bisa menyebut numpy cukup dengan np biar lebih pendek.

membuat array

- Artinya: Kita **membuat daftar angka (array)** dengan fungsi np.array().
- a dan b masing-masing berisi daftar angka [1, 2, 3] dan [4, 5, 6].

menggunakan operasi penjumlahan pada 2 array

- Artinya: Kita **menjumlahkan dua array** dengan cara a + b.

- Hasilnya: setiap posisi angka dijumlahkan $\rightarrow [1+4, 2+5, 3+6] = [5, 7, 9]$.

Indexing dan Slicing pada Array

- Artinya: Kita mengambil **bagian tertentu** dari array.
- `arr[1:3]` artinya ambil dari indeks ke-1 sampai sebelum indeks ke-3 (jadi ambil 20 dan 30).

iterasi pada array

- Artinya: Kita **melakukan perulangan** untuk melihat atau memproses setiap angka dalam array satu per satu.
- `for x in arr:` artinya ambil satu per satu dari array dan simpan sementara di `x`, lalu `print(x)` mencetaknya.

PRAKTEK 4

```
# 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 :

membuat array

- **Artinya:** Kita membuat **daftar angka** dan menyimpannya ke dalam variabel `arr`.
- Hasilnya: `arr` berisi `[1, 2, 3, 4, 5]`.

Linear Traversal ke tiap elemen arr

- **Artinya:** Kita akan **melihat satu per satu isi array** dari awal sampai akhir (disebut *linear traversal*).
- `for i in arr:` artinya: ambil satu per satu angka dari `arr` dan simpan ke `i`.
- `print(i, end=" ")` artinya: cetak angka tersebut di baris yang sama, dipisahkan dengan spasi (bukan baris baru).
- `print()` terakhir digunakan untuk membuat baris kosong setelah selesai mencetak semua angka.

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()
```

Hasil :

Reverse Traversal: 5 4 3 2 1

Penjelasan :

membuat array

- **Artinya:** Kita membuat **daftar angka** bernama arr, yang isinya adalah [1, 2, 3, 4, 5].

Reverse Traversal dari elemen akhir

- **Artinya:** Kita akan **menelusuri elemen dari belakang ke depan** (disebut *reverse traversal*).
- range(len(arr) - 1, -1, -1) artinya:
 - Mulai dari indeks terakhir (len(arr) - 1, yaitu 4)
 - Sampai ke indeks pertama (-1 artinya stop sebelum 0, jadi tetap cetak sampai indeks 0)
 - Langkah mundur (-1 berarti berjalan ke belakang)
- arr[i] mengambil elemen dari array berdasarkan indeks i.
- print(arr[i], end=" ") mencetak semua angka dari belakang ke depan dalam satu baris.

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()
```

Hasil :

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

Penjelasan :

membuat array

- **Artinya:** Kita membuat daftar angka (array) bernama arr, isinya [1, 2, 3, 4, 5].

mendeklarasikan nilai awal

- **Artinya:** Kita menyiapkan dua nilai:
 - `n = len(arr)` → menghitung jumlah elemen dalam array (`n = 5`)
 - `i = 0` → sebagai titik awal pengulangan (indeks pertama array)

Linear Traversal dengan while

- **Artinya:** Kita akan **melakukan perulangan** untuk membaca elemen array satu per satu **menggunakan perulangan while** (bukan for).
- `while i < n:` → selama nilai `i` masih kurang dari panjang array, teruskan.
- `print(arr[i], end=" ")` → tampilkan elemen ke-`i` di baris yang sama.
- `i += 1` → naikkan nilai `i` satu-satu supaya bisa pindah ke elemen berikutnya.

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)
```

Hasil :

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

Penjelasan :

membuat array

- **Artinya:** Kita membuat sebuah daftar (array) berisi angka [1, 2, 3, 4, 5] dan menyimpannya ke dalam variabel arr.

mendeklarasikan nilai awal

- **start = 0:** kita mulai dari elemen pertama (indeks 0).
- **end = len(arr) - 1:** kita mulai dari elemen terakhir (indeks 4, karena panjang array = 5).

Reverse Traversal using while loop

- **Artinya:** Kita akan menelusuri dan membalik isi array dari belakang ke depan, dengan bantuan perulangan while.

Reverse Traversal dengan while

- Perulangan while start < end: akan terus berjalan selama posisi start masih di depan end.
- arr[start], arr[end] = arr[end], arr[start] → bagian ini menukar elemen di posisi start dan end.
- start += 1 → pindah ke indeks berikutnya dari kiri.
- end -= 1 → pindah ke indeks sebelumnya dari kanan.

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))
```

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 :

membuat array

- **Artinya** : Kita bikin daftar data (array) dan isi dengan angka: 12, 16, 20, 40, 50, 70. Disimpan ke variabel bernama arr.

cetak arr sebelum penyisipan

- **Artinya** : Menampilkan isi array sebelum kita menambahkan data baru.

cetak panjang array sebelum penyisipan

- **Artinya** : Menampilkan berapa jumlah elemen yang ada di dalam array sebelum ditambah.

menyisipkan array di akhir elemen menggunakan .append()

- **Artinya** : Menambahkan data 26 ke bagian **akhir** array dengan fungsi append().

cetak arr setelah penyisipan

- **Artinya** : Menampilkan isi array setelah data 26 dimasukkan.
-

cetak panjang array setelah penyisipan

- **Artinya** : Menampilkan jumlah elemen terbaru dari array, setelah elemen baru ditambahkan.

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))
```

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 :

membuat array

- **Artinya** : Kita bikin daftar angka yang disimpan dalam variabel bernama arr.

cetak arr sebelum penyisipan

- **Artinya** : Menampilkan isi array sebelum kita tambah angka baru ke dalamnya.

cetak panjang array sebelum penyisipan

- **Artinya** : Menampilkan berapa banyak data/angka yang ada di dalam array sebelum dimodifikasi.

menyisipkan array pada tengah elemen menggunakan .insert(pos, x)

- **Artinya** : Menambahkan angka 5 ke posisi indeks 4 dalam array.
Contoh: posisi ke-4 berarti setelah elemen ke-3 dan sebelum elemen ke-4.

cetak arr setelah penyisipan

- **Artinya** : Menampilkan isi array setelah angka baru ditambahkan.

cetak panjang array setelah penyisipan

- **Artinya** : Menampilkan jumlah total elemen dalam array setelah ditambah.