UNIVERSITAS PENDIDIKAN INDONESIA

Rekayasa Perangkat Lunak

Python Programming

Array

01

Pendahuluan Array

Pendahuluan Array

- Merupakan struktur data yang digunakan untuk menyimpan sekumpulan elemen atau nilai dalam satu variabel.
- Dalam Python, struktur data yang lebih sering digunakan untuk menyimpan sekumpulan elemen adalah "list”.
- List dalam Python lebih serbaguna dan fleksibel daripada array dalam beberapa bahasa pemrograman lain.
- Perbedaan yang mencolok terletak pada cara array menyimpan nilai yang sangat berbeda dengan list.
- Dalam array, setiap elemen yang terkandung dalamnya harus memiliki tipe data yang seragam. Namun, pada list, nilai-nilai tersebut tidak diwajibkan untuk memiliki tipe data yang seragam.
- Array bukan hanya sebuah tipe data, melainkan salah satu tipe struktur data berjenis linear.
- Struktur data adalah metode untuk mengorganisir dan menyimpan informasi sehingga informasi tersebut dapat diakses dan diproses secara efisien.

Pendahuluan Array (Cont.)

- Dalam Python, kita dapat menggunakan library atau modul untuk membuat dan mengelola array atau struktur data serupa.
- Dua library yang paling umum digunakan adalah Numpy (Numerical Python) dan Array Module.
- **Library:** kumpulan kode yang telah dibuat oleh pengembang atau programmer, dan diberikan kepada pengguna lain untuk digunakan kembali dalam pengembangan perangkat lunak atau program.
- Sedangkan **modul:** berkas yang mengandung kode Python yang dapat dimanfaatkan kembali oleh programmer lain.

02
Library NumPy

Library NumPy

- NumPy adalah library yang sangat populer untuk pemrosesan numerik dan manipulasi array multidimensional.
- NumPy menyediakan struktur data inti yang disebut "ndarray" (n-dimensional array) yang sangat efisien untuk mengelola data numerik.
- Library ini juga memiliki banyak fungsi matematika yang kuat.

Instalasi NumPy

- Untuk dapat mengimpor NumPy, kita dapat menggunakan perintah `import numpy as np`.
- Namun, saat pertama kali mengimpor library NumPy kita perlu menginstall NumPy terlebih dahulu.
- Langkah-langkahnya yaitu sebagai berikut:

1. Buka terminal atau cmd, lalu install PIP

```
python3 -m pip install --upgrade pip
```

2. Apabila sudah terinstall, cek versi PIP dengan perintah sebagai berikut dan cek outputnya

```
yuliaretno@YR-MacBook-Pro ~ % pip3 --version
pip 23.2.1 from /Users/yuliaretno/Library/Python/3.9/lib/python/site-packages
/pip (python 3.9)
```

3. Install library NumPy dengan perintah `pip3 install numpy` atau `pip install numpy`

Program NumPy

- Apabila Numpy sudah berhasil di install, kita akan memulai membuat program array dengan menggunakan library Numpy.
- Berikut merupaan contoh program membuat array:

```
1  import numpy as np
2
3  arr = np.array([1, 2, 3, 4, 5])
4  print(arr)
5  print(type(arr))
6
7  list = [1, 2, 3, 4, 5]
8  print(list)
9  print(type(list))
```

✓ 0.0s

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
[1, 2, 3, 4, 5]
<class 'list'>
```

Penjelasan Program NumPy

- Program di slide sebelumnya adalah program untuk membuat array dengan menggunakan library NumPy.
- Pertama, kita mengimpor pustaka NumPy dengan perintah `import numpy as np`, sehingga kita dapat menggunakannya dengan alias "np."
- Kemudian, kita membuat sebuah array NumPy dengan perintah `np.array([1, 2, 3, 4, 5])`, yang merupakan array 1 dimensi yang berisi angka dari 1 hingga 5.
- Array ini kemudian dicetak ke layar menggunakan `print(arr)`.
- Selanjutnya, kita mencetak jenis atau tipe data dari array tersebut dengan `print(type(arr))`.
- Hasilnya adalah tampilan array NumPy dan informasi bahwa tipe data dari array tersebut adalah numpy.ndarray.

03

NumPy Array vs List

Perbedaan NumPy Array dan List

1. **Elemen**
   - **NumPy:** NumPy menggunakan array (ndarray) yang memerlukan Ape data seragam. Ini berarti semua elemen dalam array NumPy harus memiliki tipe data yang sama (misalnya, semuanya integer atau semuanya float).
   - **List:** List dalam Python tidak memerlukan tipe data seragam, hal ini memungkinkan untuk dapat memiliki elemen-elemen dengan tipe data yang berbeda dalam satu list.
2. **Kinerja**
   - **NumPy**: NumPy dirancang untuk kinerja yang tinggi dalam pemrosesan numerik. Operasi pada array NumPy dapat diimplementasikan dalam kode C di bawahnya, sehingga lebih cepat daripada list untuk komputasi numerik yang besar.
   - **List:** List adalah struktur data umum dan lebih lambat dalam operasi pemrosesan numerik dibandingkan dengan NumPy.

Perbedaan NumPy Array dan List

3. **Fungsionalitas**
   - **NumPy:** NumPy menyediakan berbagai fungsi matematik, operasi matriks, statistik, dan lainnya yang memungkinkan untuk melakukan pemrosesan data numerik yang rumit.
   - **List:** List dalam Python tidak memerlukan tipe data seragam, hal ini memungkinkan untuk dapat memiliki elemen-elemen dengan tipe data yang berbeda dalam satu list.
4. **Kemampuan Array Multidimensional**
   - **NumPy**: NumPy mendukung array multidimensional (array 2D, 3D, dll.), yang sangat berguna dalam pemrosesan data seperti gambar, ilmu data, dan ilmu komputer.
   - **List:** List hanya mendukung array 1D dan tidak memiliki dukungan bawaan untuk array multidimensional.

04

Program NumPy

Membuat Array

- Saat membuat program membuat array kita pasti harus mengimport dan mendeklarasikan NumPy.
- Objek array di NumPy disebut ndarray.
- Kita dapat membuat objek ndarray NumPy dengan menggunakan fungsi array().

```
1  import numpy as np
2
3  arr = np.array([1, 2, 3, 4, 5])
4  print(arr)
5  print(type(arr))
```

✓ 0.0s

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

Index pada Array

- Array dengan NumPy juga memiliki index yang dimulai dari 0 hingga n-1 dimana nilai n merupakan jumlah item pada suatu array.
- Kita juga bisa mengakses item pada array dengan menggunakan index dan dapat melakukan slicing pada array juga. Berikut merupakan contohnya:

```
1  import numpy as np
2
3  arr = np.array([1, 2, 3, 4, 5])
4  print(arr[1])
5  print(arr[2:5])
6  print(arr[0] + arr[4])
```

✓ 0.0s

```
2
[3 4 5]
6
```

Dimensi pada Array

- Dimensi dalam konteks array mengacu pada seberapa banyak tingkat atau tingkat yang terdapat dalam sebuah array.
- Ini berkaitan dengan sejauh mana elemen-elemen dalam array tersebut dikelompokkan atau terstruktur secara hierarkis.
- Berikut beberapa konsep dasar tentang dimensi pada array:
  1. **Array Satu Dimensi (1D):** disusun dalam satu baris atau satu kolom

```
1  import numpy as np
2
3  arr = np.array([1, 2, 3, 4, 5])
4  print(arr)
```

✓ 0.0s

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

Dimensi pada Array

2. **Array Dua Dimensi (2D):** array yang memiliki dua tingkat dimensi atau terdiri dari beberapa baris dan kolom.

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

✓ 0.0s

```
[[1 2 3]
 [4 5 6]]
```

Dimensi pada Array

3. **Array Tiga Dimensi (3D):** Array tiga dimensi adalah array yang memiliki tiga tingkat dimensi. Ini sering digunakan dalam konteks data berdimensi tinggi seperti citra 3D, volume, dan data kubus. Contoh:

```
1  import numpy as np
2
3  arr = np.array([ [[1, 2], [3, 4]], [[5, 6], [7, 8]], [[9, 10], [11, 12]]])
4  print(arr)
```

✓ 0.0s

```
[[[ 1  2]
  [ 3  4]]

 [[ 5  6]
  [ 7  8]]

 [[ 9 10]
  [11 12]]]
```

Dimensi pada Array

4. **Array N Dimensi (ND):** mengacu pada array yang memiliki lebih dari tiga tingkat dimensi. Ini sering digunakan dalam konteks data yang sangat kompleks seperti data ilmiah, jaringan saraf tiruan, atau pemrosesan gambar. Kita dapat memiliki array 4D, 5D, dan seterusnya tergantung pada kompleksitas data yang dihadapi.

Mengecek Dimensi Array

- NumPy Array menyediakan atribut `ndim` yang mengembalikan bilangan bulat yang memberi tahu berapa banyak dimensi yang dimiliki array.

```
1  import numpy as np
2
3  b = np.array([1, 2, 3, 4, 5])
4  c = np.array([[1, 2, 3], [4, 5, 6]])
5  d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
6
7  print(b.ndim)
8  print(c.ndim)
9  print(d.ndim)
```

✓ 0.0s

```
1
2
3
```

Tipe Data di Array

- Secara umum, tipe data pada Python antara lain: integer, float, string, boolean, dll.
- Namun, NumPy memiliki beberapa tipe data tambahan, dan mengacu pada tipe data dengan satu karakter. Berikut adalah penjelasannya:

  - `i` - integer
  - `b` - boolean
  - `u` - unsigned integer
  - `f` - float
  - `c` - complex float
  - `m` - timedelta
  - `M` - datetime
  - `O` - object
  - `S` - string
  - `U` - unicode string
  - `V` - fixed chunk of memory for other type (void)

Tipe Data di Array

- Untuk mengecek tipe data dari sebuah array kita bisa menggunakan properti `dtype`.
- Selain itu, properti ini juga dapat digunakan saat membuat array untuk mendefinisikan tipe data elemen pada array. Berikut merupakan contohnya:

```
1  import numpy as np
2
3  arr = np.array([1, 2, 3, 4, 5])
4  print(arr.dtype)
```

✓ 0.0s

```
int64
```

```
1  import numpy as np
2
3  arr = np.array([1, 2, 3, 4, 5], dtype="S")
4  print(arr)
5  print(arr.dtype)
```

✓ 0.0s

```
[b'1' b'2' b'3' b'4' b'5']
|S1
```

05

Studi Kasus

Studi Kasus

1. Seorang ilmuan memiliki sebuah array 1D yang isinya adalah suhu di Singapura selama 10 hari terakhir. Suhu tersebut ingin dikonversi dari Celcius menjadi Farenheit. Buatlah programnya dengan menggunakan library Numpy! (Catatan: silahkan buat data dummy untuk suhu 10 hari tersebut).

2. Seorang guru Matematika memiliki data nilai ujian sebanyak 30 orang. Guru tersebut ingin menyimpan seluruh nilai ujian tadi ke dalam sebuah array untuk kemudian diurutkan nilainya dari yang terbesar hingga terkecil, kemudian ditampilkan. Selain itu, guru tersebut ingin melihat 5 nilai terbesarnya saja. Buatlah programnya dengan menggunakan library Numpy! (Catatan: silahkan buat data dummy untuk nilai ujian 30 siswa tersebut).

UNIVERSITAS PENDIDIKAN INDONESIA

Rekayasa Perangkat Lunak

Terimakasih