

## Modul 9

### Pemrosesan Array pada Python

### Praktikum Program Komputer

### 2021

---

#### A. Pengertian

Python merupakan salah satu bahasa pemrograman yang cukup populer di kalangan praktisi data. Adapun satu jenis fungsi yang biasanya diterapkan dalam bahasa pemrograman Python yaitu Array. Salah satu ilmu dasar yang perlu kamu pahami dan kuasai dalam bidang data ialah array. Dengan mengimplementasikan array pada Python kita dapat menghasilkan sesuatu hal sederhana menjadi kompleks. Tentu saja sebelumnya harus mengetahui dasar dasar yang akan membantu dalam proses pengembangannya.

Pada artikel kali ini, kita akan mengenal lebih banyak lagi mengenai array di dalam Python. Beberapa hal yang akan kita bahas dalam artikel ini dimulai dari definisi array itu sendiri, bagaimana caranya mendefinisikan array di dalam Python, dan beberapa fungsi yang berkaitan dengan array di dalam Python. Dengan mengenal array akan sangat berguna saat melakukan Data Manipulation maupun Data Visualization dengan Python.

Array merupakan sekumpulan variabel yang memiliki tipe data yang sama dan mampu menyimpan beberapa nilai dalam satu variabel. Array merupakan tipe data terstruktur dalam pemrograman, array memungkinkan untuk menyimpan data maupun referensi objek dalam jumlah banyak dan terindeks. Sehingga sejumlah variabel dapat menggunakan nama yang sama.

#### B. Cara Mendefinisikan Array

Untuk mendefinisikan sebuah array di python bisa menggunakan keyword **nama\_array = [jumlah\_elemen]**.

Contoh mendefinisikan **suatu array**:

```
#importing "array"  
import array
```

```
>>> #inisialisasi variabel berformat array
>>> Angka = [ 1, 2, 3 ,4 ]
>>> Nama = ["Dono", "Kasino", "indro"]
>>> print Nama
>>> print Angka
```

Output:

```
['Dono', 'Kasino', 'indro']
[ 1, 2, 3 ,4 ]
```

### C. Array Konstan

Array konstan adalah array yang elemen-elemennya tidak dapat dirubah. Berikut contoh penggunaan array konstan.

#### Exercise 9.1:

```
#!C:\Python37\python

#####
# Nama file: array2.py
#####

import sys

# mendefinisikan array konstan
HARI = ('Minggu','Senin','Selasa','Rabu',
        'Kamis','Jumat','Sabtu')

def main():
    # meminta user memasukkan nomor hari
    nohari = int(input("Masukkan nomor hari [1..7]: "))

    if (nohari < 1) or (nohari > 7):
        print("Tidak ada hari ke-%s" % nohari)
        sys.exit(1)

    print("Hari ke-%d adalah %s" % (nohari, HARI[nohari-1]))

if __name__ == "__main__":
    main()
```

#### D. Array Asosiatif

Array asosiatif adalah array yang indeksnya dapat didefinisikan sendiri (biasanya berupa string). Setiap elemen yang terdapat di dalam array asosiatif berupa pasangan kunci dan nilai (sering disebut **key-value-pair**).

Berikut contoh penggunaan array asosiatif.

##### Exercise 9.2 :

```
#!C:\Python37\python

#####
# Nama file: array3.py
#####

import sys

# mendefinisikan array asosiatif
KAMUS = {
    'one': 'satu',
    'two': 'dua',
    'three': 'tiga',
    'four': 'empat',
    'five': 'lima',
    'six': 'enam',
    'seven': 'tujuh',
    'eight': 'delapan',
    'nine': 'sembilan',
    'ten': 'sepuluh'
    # ...
}

def main():
    # meminta user memasukkan kata dalam bahasa inggris
    kata = input("Masukkan kata dalam bahasa inggris: ")

    if not (kata in KAMUS.keys()):
        print("'{}' tidak ditemukan di dalam kamus ini" % kata)
        sys.exit(1)

    print("Arti kata '{}' adalah '{}'" % (kata, KAMUS[kata]))

if __name__ == "__main__":
    main()
```

## E. Array Multidimensi

Array multidimensi adalah array yang memiliki dua subskrip atau lebih. Untuk array dua dimensi, setiap elemen array yang terkandung didalamnya juga berupa array. Berikut contoh array 2 dimensi

23	11	54	38	76
14	12	20	22	21
10	13	18	17	24
35	33	39	32	29

Dalam mengakses elemen array 2 dimensi, Python menggunakan cara yang sama seperti java, yaitu dengan menggunakan bentuk umum berikut:

*NamaArray[IndexBaris][IndeksKolom]*

Dengan demikian, jika array diatas bernama A, maka setiap elemen di dalam array tersebut dapat diakses menggunakan cara berikut

A[0][0]	A[0][1]	A[0][2]	A[0][3]	A[0][4]
A[1][0]	A[1][1]	A[1][2]	A[1][3]	A[1][4]
A[2][0]	A[2][1]	A[2][2]	A[2][3]	A[2][4]
A[3][0]	A[3][1]	A[3][2]	A[3][3]	A[3][4]

Berikut ini contoh kode yang menunjukkan cara pendefinisian dan engaksesan elemen di dalam array 2 dimensi

### Exercise 9.3:

```
#I C:\Python37\python

#####
# Nama file: array4.py
#####

# mendefinisikan array dua dimensi
A = [
```



```

        [23,11,54,38,76],
        [14,12,20,22,21],
        [10,13,18,17,24],
        [35,33,39,32,29]
    ]

BARIS = 4
KOLOM = 5

def main():
    print("Isi array A:")
    # menampilkan elemen array dua dimensi
    i = 0;
    while i < BARIS:
        j = 0
        while j < KOLOM:
            print("%d " % A[i][j], end='')
            j += 1
        i += 1
        print() # ganti baris

if __name__ == "__main__":
    main()

```

#### Exercise 9.4:

```

>>> A = [
    [23,11,54,38,76],
    [14,12,20,22,21],
    [10,13,18,17,24],
    [35,33,39,32,29]
]
>>>
>>>
>>>

```

```

>>> # mengakses elemen array dua dimensi
>>> for baris in A:
    for nilai in baris:
        print("%d " % nilai, end='')
    print()

23 11 54 38 76
14 12 20 22 21
10 13 18 17 24
35 33 39 32 29
>>>

```

Berikut contoh penggunaan untuk array 3 dimensi

#### Exercise 9.5:

```

>>> A = [
    [
        [10, 20, 30],
        [40, 50, 60]
    ],
    [
        [11, 21, 31],
        [41, 51, 61]
    ]
]
>>>

```

```

>>> # mengakses elemen 10
>>> A[0][0][0]
10
>>>
>>> # mengakses elemen 50
>>> A[0][1][1]
50
>>>

```

## F. Beberapa Fungsi yang Berkaitan dengan Array di Python

### Modul Array

Objek dari kelas *array* dibuat menggunakan bentuk umum berikut:

```
array.array(KodeTipe, [nilaiInisial1, nilaiInisial2, ...])
```

Kode Tipe	Tipe C	Tipe Python	Ukuran
c	char	str (lebar 1)	1 byte
b	char	int	1 byte
B	unsigned char	int	1 byte
U	unicode char	unicode (lebar 1)	2 byte
h	short	int	2 byte
H	unsigned short	int	2 byte
i	int	int	2 byte
I	unsigned	long	2 byte
l	long	int	4 byte
L	unsigned long	long	4 byte
f	float	float	4 byte
d	double	float	8 byte

**Index:** Fungsi ini digunakan untuk mengembalikan indeks kemunculan pertama yang nilainya disebutkan dalam argument. Ini digunakan jika ada kesalahan.

**Append:** Fungsi ini digunakan untuk menambahkan nilai di akhir daftar.

**Insert:** Fungsi ini digunakan untuk menyisipkan suatu nilai pada posisi tertentu. Argumen pertama adalah indeks dari elemen sebelumnya yang dimasukkan.

**Remove:** Fungsi ini digunakan untuk menghapus nilai pertama dari daftar.

**Pop:** Fungsi ini digunakan untuk menghapus item pada posisi tertentu dalam daftar, dan mengembalikannya. Jika tidak ada indeks yang ditentukan, `a.pop ()` menghapus dan mengembalikan item terakhir dalam daftar.

**Count:** Fungsi ini digunakan untuk mengembalikan jumlah item dengan nilai yang ditentukan.

**Sort:** Fungsi ini digunakan untuk mengurutkan item dari daftar.

**Reverse:** Fungsi ini digunakan untuk mengembalikan urutan daftar.

**Extend:** Fungsi ini digunakan untuk memperpanjang daftar dengan menambahkan semua item dalam daftar yang diberikan.

**Len:** Fungsi ini digunakan untuk digunakan untuk mengembalikan nilai berupa jumlah item di daftar.

### Exercise 9.6:

```
>>> import array
>>> A = array.array('i')
>>>
```

```
>>> A.append(50)
>>> A.append(-20)
>>> A.append(30)
```

```
>>> A.insert(1, 40)
```

```
>>> A.remove(-20)
```

```
>>> # menggunakan perintah for
>>> for nilai in A:
>>>     print("%d " % nilai, end='')
```

```
>>> # menggunakan perintah while
>>> i = 0
>>> while i < len(A):
>>>     print("%d " % A[i], end='')
>>>     i += 1
```

```
>>> A.index(30)
```

### Exercise 9.7:

```
>>> # mengonversi string ke dalam array.array
>>> B = array.array('c')
>>> B.fromstring("Python")
>>>
```

```
>>> for karakter in B:
>>>     print("%c " % karakter, end='')
```

### Exercise 9.8:



```

>>> # mengonversi list ke dalam array.array
>>> li = [10, 20, 30, 40, 50]
>>> C = array.array('i')
>>> C.fromlist(li)
>>> type(C)
<type 'array.array'>
>>> for nilai in C:
    print("%d " % nilai, end='')

10 20 30 40 50
>>>

```

### Exercise 9.9:

Untuk mendefinisikan objek *array* dengan nilai awal tertentu, Anda dapat menulis kode seperti berikut:

```

>>> A = array.array('i', [100, 200, 300, 400, 500])
>>> A
array('i', [100, 200, 300, 400, 500])
>>>

```

Jika Anda ingin mengubah nilai dari elemen tertentu, tuliskan kode berikut:

```

>>> A[1] = -700 # mengubah elemen kedua
>>> A[4] = 800  # mengubah elemen kelima
>>> A
array('i', [100, -700, 300, 400, 800])
>>>

```

### Exercise 9.10:

Kelas *array* juga memiliki metode `reverse()` yang digunakan untuk membalik urutan elemen-elemen *array*. Berikut ini contoh kode yang menunjukkan penggunaan metode tersebut.

```

>>> # nilai awal (sebelum dibalik)
>>> A
array('i', [100, -700, 300, 400, 800])
>>>
>>> # membalik urutan elemen array

```

```

>>> A.reverse()
>>>
>>> # nilai akhir (setelah dibalik)
>>> A
array('i', [800, 400, 300, -700, 100])
>>>

```

### Exercise 9.11:

```

#!C:\Python37\python

#####
# Nama file: array5.py
#####

import array

# mendefinisikan fungsi untuk mengurutkan elemen array
def sort(A):
    i = 0
    while i < len(A)-1:
        j = len(A)-1
        while j >= i+1:
            if A[j] < A[j-1]:
                temp = A[j]
                A[j] = A[j-1]
                A[j-1] = temp
            j -= 1
        i += 1

def main():
    A = array.array('i', [50,10,30,40,20])
    print("Sebelum diurutkan: ")
    for nilai in A:
        print("%d " % nilai, end='')
    print("\n")

```

```
# mengurutkan array
sort(A)

print("Setelah diurutkan: ")
for nilai in A:
    print("%d " % nilai, end='')

if __name__ == "__main__":
    main()
```

Sumber :

[Belajar Python: Mengenal Array pada Bahasa Pemrograman Pytho... \(dqlab.id\)](https://dqlab.id/)

Raharjo, B. (2015). Mudah Belajar Python untuk aplikasi desktop dan web.

**Tugas!!**

**Kerjakan Exercise 9.1 – 9.11**

