



## MODUL PERKULIAHAN

# TFC251 – Praktikum Struktur Data

## Array Dalam Python

<b>Penyusun Modul</b>	:	Suamanda Ika Novichasari, M.Kom
<b>Minggu/Pertemuan</b>	:	1
<b>Sub-CPMK/Tujuan Pembelajaran</b>	:	1. Mahasiswa mampu mempraktekan konsep array pada bahasa pemrograman python
<b>Pokok Bahasan</b>	:	1. Array

**Program Studi Teknologi Informasi (S1)  
Fakultas Teknik  
Universitas Tidar  
Tahun 2024**



# Materi 1

## ARRAY DALAM PYTHON

---

### Petunjuk Praktikum :

- Cobalah semua contoh kode program yang terdapat pada semua sub bab dalam modul ini menggunakan google colab.
- Dokumentasikan kegiatan dalam bentuk laporan praktikum sesuai template yang telah ditentukan.
- Setelah selesai mempraktekan semua materi, silakan kerjakan tugas untuk persiapan praktikum pertemuan selanjutnya.

Array adalah struktur data yang merupakan kumpulan data dengan tipe data yang sama. Saat mendeklarasikan sebuah array, komputer akan memberikan slot lokasi memori yang berurutan atau berdampingan antar elemen satu dengan yang lainnya. Sehingga jumlah slot memori atau elemen array bersifat tetap sesuai jumlah saat pendeklarasian. Array juga dapat diartikan sebagai sekumpulan pasangan indeks dan nilai. Nilai dalam elemen array memiliki tipe data yang sama. Indeks array merepresentasikan alamat memori elemen array. Indeks elemen array dimulai dari 0 (nol).

10	7	5	--> <u>Nilai Element Array</u>
A[0]	A[1]	A[2]	--> <u>Alamat Element Array</u>
0	1	2	--> <u>Indeks Element Array</u>

### 1.1 Membuat Array

Array dalam *Python* dapat dibuat dengan mengimpor modul array. Fungsi `array(data_type, value_list)` digunakan untuk membuat array dengan menentukan tipe data dan daftar nilai pada argumennya.

Berikut adalah contoh kode yang membuat dua array: satu berisi bilangan bulat dan satu berisi bilangan desimal (*doubles*). Kemudian, kode tersebut mencetak isi dari masing-masing array ke konsol.

## ✓ Membuat Array

```
0d [1] import array as arr
    a = arr.array('i', [1, 2, 3])
    print("The new created array is : ", end=" ")
    for i in range(0, 3):
        print(a[i], end=" ")
    print()
    b = arr.array('d', [2.5, 3.2, 3.3])
    print("\nThe new created array is : ", end=" ")
    for i in range(0, 3):
        print(b[i], end=" ")
```

The new created array is : 1 2 3

The new created array is : 2.5 3.2 3.3

### 1.2 Menambahkan Elemen ke Array

Elemen dapat ditambahkan ke dalam Array dengan menggunakan fungsi bawaan *insert()*. Insert digunakan untuk memasukkan satu atau lebih elemen data ke dalam array. Berdasarkan kebutuhan, elemen baru dapat ditambahkan di awal, akhir, atau indeks tertentu dari array. *append()* juga digunakan untuk menambahkan nilai yang disebutkan dalam argumennya di akhir array. Berikut merupakan contoh kode menambahkan elemen array :

## Menambahkan Elemen Array

```
import array as arr
a = arr.array('i', [1, 2, 3])
print("Array before insertion : ", end=" ")
for i in range(0, 3):
    print(a[i], end=" ")
print()
a.insert(1, 4)
print("Array after insertion : ", end=" ")
for i in (a):
    print(i, end=" ")
print()
b = arr.array('d', [2.5, 3.2, 3.3])
print("Array before insertion : ", end=" ")
for i in range(0, 3):
    print(b[i], end=" ")
print()
b.append(4.4)
print("Array after insertion : ", end=" ")
for i in (b):
    print(i, end=" ")
print()
```

```
Array before insertion : 1 2 3
Array after insertion : 1 4 2 3
Array before insertion : 2.5 3.2 3.3
Array after insertion : 2.5 3.2 3.3 4.4
```

Langkah pertama mengimpor modul array sebagai arr. Kemudian, membuat array bilangan bulat bernama a dengan elemen [1, 2, 3]. Array tersebut dicetak sebelum dan setelah memasukkan bilangan bulat 4 pada indeks 1. Demikian pula, array bilangan desimal bernama b dengan elemen [2.5, 3.2, 3.3] dibuat dan dicetak sebelum dan setelah menambahkan bilangan desimal 4.4 ke dalam array.

### 1.3 Mengakses Elemen dari Array

Untuk mengakses elemen-elemen array, gunakan nomor indeks. Gunakan operator indeks [ ] untuk mengakses elemen dalam array. Indeks harus berupa bilangan bulat.

### ▼ Mengakses Elemen Array

```
import array as arr
a = arr.array('i', [1, 2, 3, 4, 5, 6])
print("Access element is: ", a[0])
print("Access element is: ", a[3])
b = arr.array('d', [2.5, 3.2, 3.3])
print("Access element is: ", b[1])
print("Access element is: ", b[2])
```

```
Access element is: 1
Access element is: 4
Access element is: 3.2
Access element is: 3.3
```

Kode ini menunjukkan penggunaan indeks untuk mengakses elemen dalam array. Ekspresi `a[0]` mengakses elemen pertama dari array `a`, yang nilainya 1. Ekspresi `a[3]` mengakses elemen keempat dari array `a`, yang nilainya 4. Demikian pula, ekspresi `b[1]` mengakses elemen kedua dari array `b`, yang nilainya 3.2, dan ekspresi `b[2]` mengakses elemen ketiga dari array `b`, yang nilainya 3.3.

#### 1.4 Menghapus Elemen Array

Elemen-elemen dapat dihapus dari array dengan menggunakan fungsi bawaan `remove()`, tetapi muncul kesalahan jika elemen tidak ada dalam set. Metode `remove()` hanya menghapus satu elemen pada satu waktu, untuk menghapus rentang elemen, iterator digunakan. Fungsi `pop()` juga dapat digunakan untuk menghapus dan mengembalikan elemen dari array, tetapi secara default hanya menghapus elemen terakhir dari array. Untuk menghapus elemen dari posisi tertentu dalam array, indeks elemen tersebut diberikan sebagai argumen untuk metode `pop()`. Metode `remove` pada List hanya akan menghapus kemunculan pertama elemen yang dicari.

### Menghapus Elemen Array

```
import array
arr = array.array('i', [1, 2, 3, 1, 5])
print("The new created array is : ", end="")
for i in range(0, 5):
    print(arr[i], end=" ")

print("\r")
print("The popped element is : ", end="")
print(arr.pop(2))
print("The array after popping is : ", end="")
for i in range(0, 4):
    print(arr[i], end=" ")

print("\r")
arr.remove(1)
print("The array after removing is : ", end="")
for i in range(0, 3):
    print(arr[i], end=" ")


```

The new created array is : 1 2 3 1 5  
The popped element is : 3  
The array after popping is : 1 2 1 5  
The array after removing is : 2 1 5

Kode ini menunjukkan bagaimana membuat, mencetak, menghapus elemen, dan mengakses elemen dari sebuah array di Python. Ini mengimpor modul array, yang digunakan untuk bekerja dengan array. Kode ini membuat array bilangan bulat dan mencetak array asli. Kemudian, satu elemen dihapus dari array dan array yang dimodifikasi dicetak. Terakhir, semua kemunculan elemen tertentu dihapus dari array dan array yang diperbarui dicetak.

## 1.5 Pemotongan Array

Dalam array Python, ada beberapa cara untuk mencetak seluruh array dengan semua elemennya, tetapi untuk mencetak rentang elemen tertentu dari array, kita menggunakan operasi Slice. Operasi Slice dilakukan pada array dengan menggunakan titik dua (:). Untuk mencetak elemen dari awal hingga suatu rentang, gunakan [:Indeks], untuk mencetak elemen dari akhir gunakan [:-Indeks], untuk mencetak elemen dari Indeks tertentu hingga akhir gunakan [Indeks:], untuk mencetak elemen dalam suatu rentang, gunakan [Indeks Awal:Indeks Akhir], dan untuk mencetak seluruh List dengan

menggunakan operasi slicing, gunakan [:]. Selanjutnya, untuk mencetak seluruh array dalam urutan terbalik, gunakan[::-1].

▼ Pemotongan Elemen Array

```
import array as arr
l = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

a = arr.array('i', l)
print("Initial Array: ")
for i in (a):
    print(i, end=" ")
Sliced_array = a[3:8]
print("\nSlicing elements in a range 3-8: ")
print(Sliced_array)
Sliced_array = a[5:]
print("\nElements sliced from 5th "
      "element till the end: ")
print(Sliced_array)
Sliced_array = a[:]
print("\nPrinting all elements using slice operation: ")
print(Sliced_array)
```

Initial Array:  
1 2 3 4 5 6 7 8 9 10  
Slicing elements in a range 3-8:  
array('i', [4, 5, 6, 7, 8])  
  
Elements sliced from 5th element till the end:  
array('i', [6, 7, 8, 9, 10])  
  
Printing all elements using slice operation:  
array('i', [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

Kode ini menggunakan slicing untuk mengekstrak elemen atau subarray dari sebuah array. Dimulai dengan array awal berisi bilangan bulat dan membuat array dari list tersebut. Kode ini melakukan slicing pada array untuk mengekstrak elemen dari indeks 3 hingga 8, dari indeks 5 hingga akhir, dan seluruh array. Array yang telah di-slice kemudian dicetak untuk mendemonstrasikan operasi slicing.

## 1.6 Mengubah Elemen dalam Array

Untuk mengubah elemen dalam array, kita hanya perlu menetapkan nilai baru ke indeks yang diinginkan untuk diubah.

### ▼ Mengubah Elemen Array

```
import array
arr = array.array('i', [1, 2, 3, 1, 2, 5])
print("Array before updation : ", end="")
for i in range(0, 6):
    print(arr[i], end=" ")

print("\n")
arr[2] = 6
print("Array after updation : ", end="")
for i in range(0, 6):
    print(arr[i], end=" ")
print()
arr[4] = 8
print("Array after updation : ", end="")
for i in range(0, 6):
    print(arr[i], end=" ")
```

Array before updation : 1 2 3 1 2 5  
Array after updation : 1 2 6 1 2 5  
Array after updation : 1 2 6 1 8 5

Contoh: Kode ini mengilustrasikan fungsionalitas memodifikasi elemen dalam sebuah array menggunakan indeks. Ini mengimpor modul array, membuat array bilangan bulat, dan mencetak array awal. Kemudian, kode ini memodifikasi dua elemen dalam array pada indeks tertentu dan mencetak array yang diperbarui. Ini bertujuan untuk menunjukkan bagaimana pengindeksan memungkinkan manipulasi dinamis dari konten array.

## 1.7 Operasi pada Array Python

### a. Menghitung Elemen dalam Array

Untuk menghitung elemen dalam sebuah array, kita perlu menggunakan metode *count*.



### ▼ Berbagai Operasi Array

```
import array
my_array = array.array('i', [1, 2, 3, 4, 2, 5, 2])

# Menghitung elemen array
count = my_array.count(2)
print("Number of occurrences of 2:", count)
```

Number of occurrences of 2: 3

Contoh: Kode ini menunjukkan bagaimana menentukan frekuensi suatu elemen tertentu dalam sebuah array. Ini mengimpor modul array, membuat array bilangan bulat, menghitung kemunculan angka 2 menggunakan metode `count()`, dan akhirnya mencetak hasilnya. Potongan kode ini secara efektif menunjukkan kemampuan untuk menganalisis distribusi elemen dalam array.

#### 1.8 Membalik Elemen dalam Array

Untuk membalik elemen-elemen dalam sebuah array, kita hanya perlu menggunakan metode `reverse`.

```
import array
my_array = array.array('i', [1, 2, 3, 4, 2, 5, 2])

# Menghitung elemen array
count = my_array.count(2)
print("Number of occurrences of 2:", count)

# Membalik Elemen Array
print("Original array:", *my_array)
my_array.reverse()
print("Reversed array:", *my_array)
```

Number of occurrences of 2: 3  
Original array: 1 2 3 4 2 5 2  
Reversed array: 2 5 2 4 3 2 1

Contoh: Kode yang disajikan menggambarkan fungsionalitas membalik urutan elemen dalam sebuah array dengan menggunakan metode `reverse()`. Ini mengimpor modul array, membuat array bilangan bulat, mencetak array asli, membalik urutan elemen menggunakan `reverse()`, dan kemudian mencetak array yang telah dibalik. Ini

secara efektif mengilustrasikan kemampuan untuk memodifikasi susunan elemen dalam sebuah array.

### 1.9 Extending Elemen dalam Array

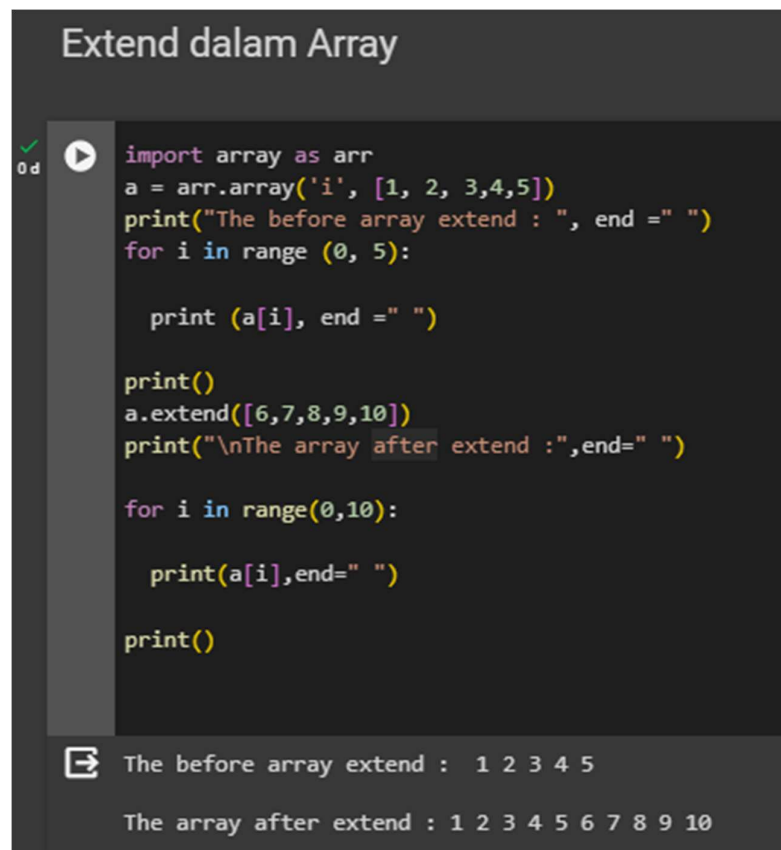
Dalam Python, array digunakan untuk menyimpan beberapa nilai atau elemen dengan tipe data yang sama dalam satu variabel. Fungsi *extend()* digunakan untuk menambahkan item dari iterable ke bagian akhir array. Dengan kata lain, metode ini digunakan untuk menambahkan array nilai ke bagian akhir array yang ada atau yang diberikan.

Syntax dari metode *extend()* adalah sebagai berikut :

```
list.extend(iterable)
```

Melalui kode tersebut semua elemen dari iterable ditambahkan ke bagian akhir dari list1.

Contoh 1:



The screenshot shows a code editor with a dark background. At the top, the title "Extend dalam Array" is displayed. Below the title, there is a code block containing Python code. The code imports the array module as 'arr', creates an array 'a' with type 'i' and values [1, 2, 3, 4, 5], prints the array before extension, and then extends it with the list [6, 7, 8, 9, 10]. Finally, it prints the array after extension. The output at the bottom shows the array before and after extension.

```
import array as arr
a = arr.array('i', [1, 2, 3, 4, 5])
print("The before array extend : ", end=" ")
for i in range (0, 5):
    print (a[i], end=" ")

print()
a.extend([6, 7, 8, 9, 10])
print("\nThe array after extend :", end=" ")

for i in range(0, 10):
    print(a[i], end=" ")

print()
```

The before array extend : 1 2 3 4 5

The array after extend : 1 2 3 4 5 6 7 8 9 10

Kode yang diberikan mendemonstrasikan kemampuan untuk memperluas sebuah array untuk menyertakan elemen tambahan. Ini mengimpor modul array menggunakan alias, membuat array bilangan bulat, mencetak array sebelum dilakukan ekstensi, memperluas array menggunakan metode `extend()`, dan akhirnya mencetak array yang sudah diperluas. Ini dengan singkat mengilustrasikan kemampuan untuk menambahkan elemen ke dalam struktur array yang sudah ada.

Contoh 2:

```
import array as arr
a=arr.array('i',[1,2,3,4,5,6])
print("The Before extend array is :",end=" ")

for i in range(0,6):
    print(a[i],end=" ")
print()
a.extend([7,8,9,10,11,12])
print("\nThe After extend array is :",end=" ")

for i in range(0,12):
    print(a[i],end=" ")
print()
b = arr.array('d', [2.1,2.2,2.3,2.4,2.5,2.6])
print("\nThe before extend array is :",end=" ")

for i in range(0,6):
    print(b[i],end=" ")
print()
b.extend([2.6,2.7,2.8,2.9])
print("\nThe after extend array is :",end=" ")

for i in range(0,9+1):
    print(b[i],end=" ")
print()
```

The Before extend array is : 1 2 3 4 5 6

The After extend array is : 1 2 3 4 5 6 7 8 9 10 11 12

The before extend array is : 2.1 2.2 2.3 2.4 2.5 2.6

The after extend array is : 2.1 2.2 2.3 2.4 2.5 2.6 2.6 2.7 2.8 2.9

Kode yang disediakan menunjukkan kemampuan untuk memperluas array dengan berbagai jenis data, termasuk bilangan bulat dan desimal. Ini menggunakan modul array, membuat array dari kedua jenis data tersebut, dan memperluasnya menggunakan metode `extend()`. Array-array tersebut kemudian dicetak sebelum dan setelah dilakukan ekstensi untuk mengilustrasikan perubahan. Ini dengan efektif

memperlihatkan kemampuan untuk menambahkan elemen ke dalam array dengan representasi data yang berbeda.

## TUGAS 1

---

Rangkumlah materi tentang stack !

Tugas dikumpulkan pada elita maksimal 1 jam sebelum pertemuan selanjutnya.