

LAPORAN AKHIR MODUL 1

Mata Kuliah : Praktikum Struktur Data

Semester : II (Dua)

Rombel : 04 (Empat)



Penyusun :

Nama Mahasiswa : Oktario Mufti Yudha

NPM : 2320506044

PROGRAM STUDI S1 TEKNOLOGI INFORMASI

UNIVERSITAS TIDAR

Genap 2023/2024

A. Dasar Teori

a) Prinsip Kerja

1. Interpreted language .

Python merupakan bahasa pemrograman yang diinterpretasi bukan di compile

2. High-Level Language.

Python merupakan bahasa pemrograman tingkat tinggi. Dimana semakin tinggi tingkatan Bahasa pemrograman maka akan semakin mudah untuk dipahami oleh para programmer

3. Modular.

Code yang menggunakan bahasa pemrograman python dapat di pisahkan menjadi modul modul kecil yang nantinya dapat kita gunakan Kembali

4. Dynamic typing.

Kita tidak perlu menulis tipe data Ketika ingin membuat variable karena python sudah menentukan sendiri tipe data tersebut sesuai dengan nilai yang kita masukan.

5. Open Source

Python merupakan bahasa pemrograman yang memiliki sumber terbuka. Jadi kita dapat melihat memodif atau menghapus sumber dari bahasa python itu sendiri.

b) Dasar Teori

Array adalah struktur data yang digunakan untuk menyimpan kumpulan elemen data yang serupa, yang disusun dalam urutan tertentu dan diakses melalui indeks numerik. Dasar teori tentang array meliputi konsep, operasi dasar, penggunaan, dan kompleksitas.

B. Uraian Kode Program

a) Membuat Array

```

# Membuat Array
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('the new created array is : ', end = ' ')
for i in range(0, 3):
    print(b[i], end = ' ')
print()

```

Gambar 2.1: Membuat Array

- `import array as arr`: Mengimpor modul array dan memberikan alias `arr` untuk mempermudah penggunaan.
- `a = arr.array('i', [1, 2, 3])`: Membuat sebuah array bernama `a` dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3].
- `print('the new created array is : ', end = ' ')`: Mencetak teks "the new created array is :" tanpa baris baru sebagai akhiran.
- `for i in range(0, 3):`: Melakukan loop dari 0 hingga 2.
- `print(a[i], end = ' ')`: Mencetak setiap elemen dari array `a`, yaitu 1, 2, dan 3, dengan spasi sebagai pemisah.
- `print()`: Mencetak baris baru, sehingga hasil cetakan selanjutnya akan berada di baris baru.
- `b = arr.array('d', [2.5, 3.2, 3.3])`: Membuat array baru bernama `b` dengan tipe data double ('d') dan elemen-elemen [2.5, 3.2, 3.3].
- `print('the new created array is : ', end = ' ')`: Mencetak teks "the new created array is :" tanpa baris baru sebagai akhiran.
- `for i in range(0, 3):`: Melakukan loop dari 0 hingga 2.
- `print(b[i], end = ' ')`: Mencetak setiap elemen dari array `b`, yaitu 2.5, 3.2, dan 3.3, dengan spasi sebagai pemisah.

b) Menambahkan Elemen ke Array

```

# Menambahkan Elemen ke Array
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()
a.insert(1, 4)
print('array after insertion is : ', end = ' ')
for i in (a):
    print(i, end = ' ')
print()

b = arr.array('d', [2.5, 3.2, 3.3])
print('the new created array is : ', end = ' ')
for i in range(0, 3):
    print(b[i], end = ' ')
print()
b.append(4.4)
print('array after inserttion is : ', end = ' ')
for i in (a):
    print(i, end = ' ')
print()

```

Gambar 2.2: Menambahkan Elemen ke Array

- `import array as arr`: Mengimpor modul array dan memberikan alias arr untuk mempermudah penggunaan.
- `a = arr.array('i', [1, 2, 3])`: Membuat sebuah array bernama a dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3].
- `print('the new created array is : ', end = ' ')`: Mencetak teks "the new created array is : " tanpa baris baru sebagai akhiran.
- `for i in range(0, 3)::` Melakukan loop dari 0 hingga 2.
- `print(a[i], end = ' ')`: Mencetak setiap elemen dari array a, yaitu 1, 2, dan 3, dengan spasi sebagai pemisah.
- `print()`: Mencetak baris baru.
- `a.insert(1, 4)`: Memasukkan nilai 4 ke indeks ke-1 dalam array a.
- `print('array after insertion is : ', end = ' ')`: Mencetak teks "array after insertion is : " tanpa baris baru sebagai akhiran.
- `for i in (a)::` Melakukan loop untuk setiap elemen dalam array a.
- `print(i, end = ' ')`: Mencetak setiap elemen dalam array a, termasuk nilai yang baru dimasukkan (1, 4, 2, 3), dengan spasi sebagai pemisah.
- `print()`: Mencetak baris baru.
- `b = arr.array('d', [2.5, 3.2, 3.3])`: Membuat array baru bernama b dengan tipe data double ('d') dan elemen-elemen [2.5, 3.2, 3.3].

- `print('the new created array is : ', end = ' ')`: Mencetak teks "the new created array is :" tanpa baris baru sebagai akhiran.
- `for i in range(0, 3)::` Melakukan loop dari 0 hingga 2.
- `print(b[i], end = ' ')`: Mencetak setiap elemen dari array b, yaitu 2.5, 3.2, dan 3.3, dengan spasi sebagai pemisah.
- `print()`: Mencetak baris baru.
- `b.append(4.4)`: Menambahkan nilai 4.4 ke akhir array b.
- `print('array after inserttion is : ', end = ' ')`: Mencetak teks "array after insertion is :" tanpa baris baru sebagai akhiran.
- `for i in (a)::` Melakukan loop untuk setiap elemen dalam array a.
- `print(i, end = ' ')`: Mencetak setiap elemen dalam array a, karena kode ini seharusnya mencetak array b, ada kesalahan penulisan, sehingga hasilnya adalah (1, 4, 2, 3, 4.4), dengan spasi sebagai pemisah.
- `print()`: Mencetak baris baru.

c) Mengakses Elemen dari Array

```
# Mengakses Elemen dari 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])
```

Gambar 2.3: Mengakses Elemen dari Array

- `import array as arr`: Mengimpor modul array dan memberikan alias arr untuk mempermudah penggunaan.
- `a = arr.array('i', [1, 2, 3, 4, 5, 6])`: Membuat sebuah array bernama a dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3, 4, 5, 6].
- `print("Access element is: ", a[0])`: Mencetak teks "Access element is: " diikuti dengan nilai elemen pertama dari array a, yaitu 1.

- `print("Access element is: ", a[3]):` Mencetak teks "Access element is: " diikuti dengan nilai elemen keempat dari array a, yaitu 4.
- `b = arr.array('d', [2.5, 3.2, 3.3]):` Membuat array baru bernama b dengan tipe data double ('d') dan elemen-elemen [2.5, 3.2, 3.3].
- `print("Access element is: ", b[1]):` Mencetak teks "Access element is: " diikuti dengan nilai elemen kedua dari array b, yaitu 3.2.
- `print("Access element is: ", b[2]):` Mencetak teks "Access element is: " diikuti dengan nilai elemen ketiga dari array b, yaitu 3.3.

d) Menghapus Elemen Array

```
# 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=" ")
```

Gambar 2.4: Menghapus Elemen Array

- `import array:` Mengimpor modul array tanpa memberikan alias.
- `arr = array.array('i', [1, 2, 3, 1, 5]):` Membuat sebuah array Bernama arr dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3, 1, 5].
- `print("The new created array is : ", end=" ")`: Mencetak teks "The new created array is : " tanpa baris baru sebagai akhiran.
- `for i in range(0, 5)::` Melakukan loop dari 0 hingga 4.
- `print(arr[i], end=" ")`: Mencetak setiap elemen dari array arr, yaitu 1, 2, 3, 1, 5, dengan spasi sebagai pemisah.
- `print("\r")`: Mencetak baris baru.
- `print("The popped element is : ", end=" ")`: Mencetak teks "The popped element is : " tanpa baris baru sebagai akhiran.

- `print(arr.pop(2))`: Memanggil metode `pop(2)` pada array `arr`, yang menghapus dan mengembalikan elemen pada indeks ke-2 dari array tersebut. Elemen yang dihapus adalah 3.
- `print("The array after popping is : ", end=" ")`: Mencetak teks "The array after popping is :" tanpa baris baru sebagai akhiran.
- `for i in range(0, 4)::` Melakukan loop dari 0 hingga 3.
- `print(arr[i], end=" ")`: Mencetak setiap elemen dari array `arr` setelah operasi popping, yaitu 1, 2, 1, 5, dengan spasi sebagai pemisah.
- `print("\r")`: Mencetak baris baru.
- `arr.remove(1)`: Menghapus nilai pertama yang bernilai 1 dari array `arr`.
- `print("The array after removing is : ", end=" ")`: Mencetak teks "The array after removing is :" tanpa baris baru sebagai akhiran.
- `for i in range(0, 3)::` Melakukan loop dari 0 hingga 2.
- `print(arr[i], end=" ")`: Mencetak setiap elemen dari array `arr` setelah operasi penghapusan, yaitu 2, 1, 5, dengan spasi sebagai pemisah.

e) Pemotongan Array

```
# Pemotongan 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)
```

Gambar 2.5: Pemotongan Array

- `import array as arr`: Mengimpor modul array dan memberikan alias `arr` untuk mempermudah penggunaan.
- `l = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`: Membuat list `l` yang berisi 10 elemen.
- `a = arr.array('i', l)`: Membuat array bernama `a` dengan tipe data integer ('i') dan elemen-elemen dari list `l`.

- `print("Initial Array : ")`: Mencetak teks "Initial Array : ".
- `for i in (a)::`: Melakukan loop untuk setiap elemen dalam array a.
- `print(i, end=" ")`: Mencetak setiap elemen dari array a dengan spasi sebagai pemisah.
- `Sliced_array = a[3:8]`: Memotong array a dari indeks ke-3 hingga indeks ke-7 (indeks terakhir tidak termasuk), menghasilkan potongan array dengan elemen-elemen [4, 5, 6, 7, 8].
- `print("\nSlicing elements in a range 3-8 : ")`: Mencetak teks "Slicing elements in a range 3-8 :" dengan baris baru.
- `print(Sliced_array)`: Mencetak potongan array yang dihasilkan.
- `Sliced_array = a[5:]`: Memotong array a mulai dari indeks ke-5 hingga akhir array.
- `print("\nElements sliced from 5th " "element till the end : ")`: Mencetak teks "Elements sliced from 5th element till the end :" dengan baris baru.
- `print(Sliced_array)`: Mencetak potongan array yang dihasilkan.
- `Sliced_array = a[:]`: Memotong seluruh array a, menghasilkan salinan lengkap array.
- `print("\nPrinting all elements using slice operation : ")`: Mencetak teks "Printing all elements using slice operation :" dengan baris baru.
- `print(Sliced_array)`: Mencetak potongan array yang dihasilkan.

f) Mengubah Elemen dalam Array

```
# Mengubah Elemen dalam 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=" ")
```

Gambar 2.6: Mengubah Elemen Array

- `import array`: Mengimpor modul array tanpa memberikan alias.
- `arr = array.array('i', [1, 2, 3, 1, 2, 5])`: Membuat sebuah array bernama arr dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3, 1, 2, 5].
- `print("Array before updation : ", end=" ")`: Mencetak teks "Array before updation :" tanpa baris baru sebagai akhiran.
- `for i in range(0, 6)::` Melakukan loop dari 0 hingga 5.
- `print(arr[i], end=" ")`: Mencetak setiap elemen dari array arr dengan spasi sebagai pemisah.
- `print("\n")`: Mencetak baris baru.
- `arr[2] = 6`: Memperbarui nilai elemen kedua dari array arr menjadi 6.
- `print("Array after updation : ", end=" ")`: Mencetak teks "Array after updation :" tanpa baris baru sebagai akhiran.
- `for i in range(0, 6)::` Melakukan loop dari 0 hingga 5.
- `print(arr[i], end=" ")`: Mencetak setiap elemen dari array arr setelah pembaruan.
- `arr[4] = 8`: Memperbarui nilai elemen keempat dari array arr menjadi 8.
- `print("Array after updation : ", end=" ")`: Mencetak teks "Array after updation :" tanpa baris baru sebagai akhiran.
- `for i in range(0, 6)::` Melakukan loop dari 0 hingga 5.
- `print(arr[i], end=" ")`: Mencetak setiap elemen dari array arr setelah pembaruan.

g) Menghitung Elemen Array

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

count = my_array.count(2)
print("Number of occurrences of 2 : ", count)
```

Gambar 2.7: Menghitung Elemen Array

- `Import array`: Mengimpor modul array untuk digunakan dalam skrip Python.
- `my_array = array.array('i', [1, 2, 3, 4, 2, 5, 2])`: Membuat array dengan nama `my_array` yang berisi bilangan bulat (tipe data 'i') [1, 2, 3, 4, 2, 5, 2].
- `count = my_array.count(2)`: Menghitung jumlah kemunculan nilai 2 dalam array `my_array` menggunakan metode `count()`. Hasilnya disimpan dalam variabel `count`.
- `print("Number of occurrences of 2 : ", count)`: Mencetak jumlah kemunculan nilai 2 dalam array `my_array` dengan teks "Number of occurrences of 2 : " diikuti dengan nilai variabel `count`.

h) Membalik Elemen Array

```
# Membalik Elemen dalam Array
import array
my_array = array.array('i', [1, 2, 3, 4, 2, 5, 2])

count = my_array.count(2)
print("Number of occurrences of 2 : ", count)

print("Original array : ", *my_array)
my_array.reverse()
print("Reversed array : ", *my_array)
```

Gambar 2.8: Membalikan Elemen Array

- `import array`: Mengimpor modul 'array' untuk menggunakan tipe data array
- `my_array = array.array('i', [1, 2, 3, 4, 2, 5, 2])` : Membuat array 'my_array' dengan tipe data 'i' (integer) dan nilai-nilai yang diberikan
- `count = my_array.count(2)` : Menghitung berapa kali nilai 2 muncul dalam array 'my_array' dan menyimpan hasilnya dalam variabel 'count'
- `print("Number of occurrences of 2 : ", count)`: Mencetak jumlah kemunculan nilai 2 dalam array 'my_array'
- `print("Original array : ", *my_array)`: Mencetak pesan dan nilai-nilai asli dari array 'my_array'
- `my_array.reverse()`: Membalik urutan elemen dalam array 'my_array'

- `print("Reversed array : ", *my_array):` Mencetak pesan dan nilai-nilai dari array 'my_array' yang sudah dibalik urutannya

i) Extending Elemen dalam Array 1

```
# Extending Elemen dalam Array 1
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()
```

Gambar 2.9: Extending Elemen dalam Array 1

- `import array as arr:` Mengimpor modul 'array' dan memberi alias 'arr'
- `a = arr.array('i', [1, 2, 3, 4, 5]):` Membuat array 'a' dengan tipe data 'i' (integer) dan nilai-nilai yang diberikan
- `print("The before array extend : ", end=" "):` Mencetak pesan
- `for i in range(0, 5):` Looping untuk mencetak nilai-nilai array 'a' sebelum dilakukan operasi extend
- `print(a[i], end=" "):` Mencetak nilai array 'a' pada indeks ke-i, dengan spasi sebagai pemisah
- `print():` Mencetak baris baru
- `a.extend([6, 7, 8, 9, 10]):` Menambahkan nilai 6, 7, 8, 9, dan 10 ke dalam array 'a'
- `print("\nThe array after extend : ", end=""):` Mencetak pesan
- `for i in range(0, 10):` Looping untuk mencetak nilai-nilai array 'a' setelah dilakukan operasi extend
- `print(a[i], end=" "):` Mencetak nilai array 'a' pada indeks ke-i, dengan spasi sebagai pemisah

- `print()`: Mencetak baris baru

j) Extending Elemen dalam Array 2

```
# Extending Elemen dalam Array 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([17,8,9,10,11,12])
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()
```

Gambar 2.10: Extending Elemen Array 2

- `import array as arr`: Mengimpor modul 'array' dan memberi alias 'arr'
- `a = arr.array('i', [1, 2, 3, 4, 5, 6])`: Membuat array 'a' dengan tipe data 'i' (integer) dan nilai-nilai yang diberikan
- `print("The Before extend array is : ", end=" ")`: Mencetak pesan
- `for i in range(0, 6)`: Looping untuk mencetak nilai-nilai array 'a' sebelum dilakukan operasi extend
- `print(a[i], end=" ")`: Mencetak nilai array 'a' pada indeks ke-i, dengan spasi sebagai pemisah
- `print()`: Mencetak baris baru
- `a.extend([17, 8, 9, 10, 11, 12])`: Menambahkan nilai 17, 8, 9, 10, 11, dan 12 ke dalam array 'a'
- `print("\nThe After extend array is : ", end=" ")`: Mencetak pesan
- `for i in range(0, 12)`: Looping untuk mencetak nilai-nilai array 'a' setelah dilakukan operasi extend

- `print(a[i], end=" ")`: Mencetak nilai array 'a' pada indeks ke-i, dengan spasi sebagai pemisah
- `print()`: Mencetak baris baru
- `b = arr.array('d', [2.1, 2.2, 2.3, 2.4, 2.5, 2.6])`: Membuat array 'b' dengan tipe data 'd' (double) dan nilai-nilai yang diberikan
- `print("\nThe before extend array is : ", end=" ")`: Mencetak pesan
- `for i in range(0, 6)`: Looping untuk mencetak nilai-nilai array 'b' sebelum dilakukan operasi extend
- `print(b[i], end=" ")`: Mencetak nilai array 'b' pada indeks ke-i, dengan spasi sebagai pemisah
- `print()`: Mencetak baris baru
- `b.extend([2.6, 2.7, 2.8, 2.9])`: Menambahkan nilai 2.6, 2.7, 2.8, dan 2.9 ke dalam array 'b'
- `print("\nThe after extend array is : ", end=" ")`: Mencetak pesan
- `for i in range(0, 9+1)`: Looping untuk mencetak nilai-nilai array 'b' setelah dilakukan operasi extend
- `print(b[i], end=" ")`: Mencetak nilai array 'b' pada indeks ke-i, dengan spasi sebagai pemisah
- `print()`: Mencetak baris baru

C. Kesimpulan

Kode-kode di atas menggunakan modul array dalam Python untuk membuat dan memanipulasi array. Dalam kode-kode tersebut, digunakan fungsi `array.array('i', [list_of_integers])` untuk membuat array dengan tipe data integer. Kemudian, metode `count()` digunakan untuk menghitung berapa kali suatu nilai muncul dalam array, sedangkan metode `reverse()` digunakan untuk membalikkan urutan elemen dalam array. Untuk mencetak nilai-nilai array, digunakan fungsi `print()`, dan operator `*` digunakan untuk melewati elemen-elemen array sebagai argumen terpisah ke fungsi `print()`.