LAPORAN AKHIR MODUL 1

Mata Kuliah : Praktikum Struktur Data

Semester : II (Dua)

Rombel : 03 (Tiga)

Penyusun :

Nama Mahasiswa : Restu Wibisono

NPM : 2340506061

PROGRAM STUDI S1 TEKNOLOGI INFORMASI

UNIVERSITAS TIDAR

Genap 2023/2024

A. Dasar Teori

1. Prinsip Kerja

Python didesain dengan layout kode yang bersih dan mudah dibaca, layout tersebut menggunakan spasi dan tab yang membuat kode terlihat seperti teks pada umumnya. Python adalah bahasa pemrograman interpretatif, yang berarti kode dieksekusi baris demi baris. Bahasa pemrograman ini merupakan bahasa tingkat tinggi yang memungkinkan operasi atau kode kompleks diungkapkan dalam kode sederhana.

Python memiliki banyak perpustakaan yang mencakup banyak fungsi dan modul siap pakai, membuat pengembangan perangkat lunak menjadi lebih mudah dan cepat. Python dapat digunakan di berbagai platform tanpa modifikasi besar. Bahasa pemrograman ini bersifat open source yang artinya kita dapat mendownload, mengganti dan mendistribusikan kodenya, python dapat digunakan untuk berbagai keperluan seperti pengembangan web, ilmu data, kecerdasan buatan, dll.

2. Dasar Teori

Array adalah struktur data yang merupakan kumpulan data dengan tipe yang sama. Saat membuat sebuah array, komputer akan memberikan selot memori yang berurutan antar elemen dengan yang lain. Jumlah 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 dara yang sama. Indeks array mempresentasikan alamat memori elemen. Indeks elemen array dimulai dari 0 (nol).

Dalam Praktikum ini, mempelajari array akan melibatkan pemahaman mendasar tentang konsep dan penggunaan array di dalam pemrograman. Array digunakan untuk menyimpan kumpulan elemen-elemen yang serupa, seperti string atau angka, dalam variabel yang sama. Saat membuat array, variabel array menyediakan elemen yang bisa disimpan di dalamnya. Untuk menambahkan elemen dalam array, bisa menggunakan metode ‘append()’ atau mengakses indeks tertentu dan mengganti nilainya. Untuk menghapus elemen dalam array, bisa menggunakan ‘remove()’ atau ‘pop()’ untuk menghapus di indeks tertentu atau dapat digunakan untuk pemotongan indeks awal dan akhir. Mengubah elemen di dalam array bisa menggunakan ‘reverse()’.

B. Uraian Kode Program

1. Membuat Array

Array dalam Python dapat dibuat dengan mengimpor modul array. Fungsi array digunakan untuk membuat array dengan menentukan tipe data dan daftar nilai pada argumen.

```
import array as arr
a = arr.array('i', [1,2,3,4,5])
print("The new created array is : ", end="")
for i in range(0, 5):
  print(a[i], end="")
print()
b = arr.array('d', [1.2, 2.5, 3.2, 3.3])
print("\nThe new created array is : ", end=" ")
for i in range(0,4):
  print(b[i], end=" ")
```

```
The new created array is : 12345

The new created array is :  1.2 2.5 3.2 3.3
```

(Gambar 2.1)

- Pertama import modul array dengan perintah ‘import array as arr’ sehingga modul dapat diakses menggunakan alias ‘arr’.
- Selanjutnya program membuat a array ‘a’ dengan memakai fungsi ‘array()’ dari modul array. Array ‘a’ berisikan bilangan bulat dengan tipe data ‘i’ dengan elemen [1,2,3,4,5]
- Lalu mencetak pesan “The new created array is :” tanpa membuat barisan baru setelahnya.
- Program akan mencetak array ‘a’ dengan melakukan iterasi melalui setiap elemen menggunakan loop ‘for’.
- Selanjutnya, membuat array ‘b’ dengan tipe data ‘d’ dengan elemen [1.2,2.5,3.2,3.3].
- Lalu akan mencetak string “The new created array is : " tanpa membuat barisan baru setelahnya.
- Array ‘b’ menggunakan proses yang sama seperti array ‘a’

2. Menambahkan Elemen ke Array

Elemen dapat ditambahkan ke dalam array dengan menggunakan fungsi bahwa insert(). Insert berfungsi sebagai 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 argumen di akhir array.

```
import array as arr
a = arr.array('i', [1, 2, 3, 4, 5])
print("Array sebelum disisipkan : ", end=" ")
for i in range (0, 5):
  print(a[i], end=" ")
print()
a.insert(3, 7)
print("Array setelah disisipkan : ", end=" ")
for i in (a):
  print(i, end=" ")
print()
b = arr.array('d', [1.2, 2.5, 3.2, 3.3])
print("Array before insertion   : ", end=" ")
for i in range(0, 4):
  print(b[i], end=" ")
print()
b.append(5.5)
print("Array after insettion    : ", end=" ")
for i in (b):
  print(i, end=" ")
print()
```

```
Array sebelum disisipkan :  1 2 3 4 5
Array setelah disisipkan :  1 2 3 7 4 5
Array before insertion   :  1.2 2.5 3.2 3.3
Array after insettion    :  1.2 2.5 3.2 3.3 5.5
```

(Gambar 2.2)

- Program dimulai dengan mengimport modul ‘array’ dan memberikan alias ‘arr’ sehingga modul array bisa dipanggil dengan alias ‘arr’
- Kemudian membuat array dengan nama ‘a’ memakai fungsi ‘array()’ dari modul ‘arr’.
- Array tersebut memiliki tipe data ‘i’ dan elemen [1,2,3,4,5].
- Program akan mencetak string “Array sebelum disisipkan :” tanpa membuat baris baru setelahnya.
- Selanjutnya menggunakan loop ‘for’, untuk mencetak setiap elemen array ‘a’ secara berurutan.
- Metode ‘insert()’ digunakan untuk menyisipkan nilai 7 ke array ‘a’ pada indeks ke-3.
- Pada baris selanjutnya akan mencetak string “Array setelah disisipkan :” tanpa membuat baris baru setelahnya.
- Loop ‘for’ akan mencetak setiap elemen array ‘a’ setelah dilakukan penyisipan.
- Membuat array dengan nama ‘b’ menggunakan fungsi ‘array()’ dari modul alias ‘arr’.
- Array tersebut diberikan tipe data ‘d’ dan elemen [1.2,2.5,3.2,3.3].
- Pada baris selanjutknya akan mencetak string “Array before insertion :”tanpa membuat baris baru setelahnya.
- Selanjutnya menggunakan ‘append()’ untuk menambah nilai 5.5 ke akhir array.
- Lalu akan mencetak string “Array after insertion :” tanpa baris baru selanjutnya.
- Terakhir memakai loop ‘for;, baris ini akan mencetak setiap elemen array ‘b”.

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.

```
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
```

(Gambar 2.3)

- Baris pertama mengimport modul ‘array’ dan memberikan alias ‘arr’.
- Baris kedua membuat array dengan nama ‘a’ menggunakan fungsi ‘array()’ dari modul ‘arr’ dengan tipe data ‘i’ dan elemen [1,2,3,4,5,6]
- Baris ini mencetak string “Access element is :” diikuti dengan akses elemen keempat dari array a, yaitu a[3].
- Selanjutnya membuat array dengan nama ‘b’ menggunakan ‘array()’ dari modul ‘arr’ dan memiliki tipe data ‘d’ dengan elemen [2.5, 3.2, 3.3]
- Kemudian program mencetak string "Access element is: " diikuti dengan akses elemen kedua dari array b, yaitu b[1].
- Terakhir akan mencetak "Access element is: " dengan elemen ke 3 yaitu ‘b[2]’

4. Menghapus Elemen Array

Elemen-elemen dapat dihapus dari array dengan mengunakan 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 bisa berfungsi untuk menghapus dan mengembalikan elemen dalam array, akan tetapi secara bawaan hanya untuk menghapus elemen terakhir dalam array. Untuk menghapus posisi tertentu menggunakan pop(). Metode remove pada List hanya menghapus kemunculan pertama elemen yang dicari.

```
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: 1215
The array after removing is : 215
```

(Gambar 2.4)

- Import modul ‘array’.
- Membuat variabel dengan nama ‘arr’ menggunakan fungsi ‘array()’ dari modul ‘array’ fsn mrmiliki tipe data ‘i’ dan elemen [1,2,3,4,5,]
- Selanjutnya program akan mencetak string “The nwe created array is:”, kemudian menggunakan loop ‘for’, dan akan mencetak setiap elemen array ‘arr’
- ‘/r’ akan membuat baris baru.
- Lalu program akan mencetak string “The array after popping: “ lalu akan menghapus dan mencetak elemen array ‘arr’ di indeks ke-2 memakai metode ‘pop()’.
- Pada baris ‘arr.remove()’ berfungsi untuk menghapus kemunculan pertama dengan nilai 1 dari array ‘arr’
- Terkahir akan mencetak string “Ther array after removing is :” lalu akan menggunakan loop ‘for’ dan mencetak elemen array ‘arr’ setelah operasi penghapusan.

5. Pemotongan Array

Dalam array Python, ada beberapa cara untuk mencetak seluruh array dengan semua elemennya, tetapi untuk bisa mencetak seluruh elemen dengan rentang tertentu bisa menggunakan operasi Slice. Operasi Slice ini dilakukan dalam array menggunakan titik dua (:). Untuk membuat elemen dari awal hingga suatu rentang, bisa menggunakan [:indeks], untuk membuat elemen dari akhir bisa memakai [indeks:], untuk mencetak elemen dengan rentang bisa menggunakan [indeks awal:indeks akhir], dan untuk mencetak seluruh List bisa

dengan memakai operasi slicing, menggunakan [:], lalu untuk mencetak array dengan urutan terbalik bisa menggunakan [::-1]

```
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])
```

(Gambar 2.5)

- Import modul ‘array’ dan berikan alias ‘arr’
- Membuat list ‘l’ yang berisikan angka dari 1 sampai 10.
- Lalu membuat array dengan nama ‘a’ memakai fungsi ‘array()’ dari modul ‘arr’ dengan tipe data ‘i’ dan elemen array ‘a’.
- Program akan mencetak sting “Inital Array:” dan akan menggunakan loop ‘for’ serta mencetak setiap elemen array ‘a’.
- Baris selanjutnya motong array ‘a’ dari indeks 3 sampai 7 menggunakan sliching dan menetapkan ke variabel ‘Sliced_array’.
- Selanjutnya akan memotong aray ‘a’ dari indeks 5 sampai array terakhir menggunakan sliching dan menetapkannya.
- Baris terakhir akan melakukan sliching pada seluruh array ‘a’ dan akan menghasilkan salinan lengkap dari array.

6. Mengubah Elemen dalam Array

Untuk mengubah elemen di dalam array, hanya diperlukan untuk menetapkan nilai baru ke dalam indeks yang ingin diubah.

```
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("\r")
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
```

(Gambar 2.6)

- Import modul ‘array’
- Buat variabel array dengan nama ‘arr’ menggunakan fungsi ‘array()’ dari modul ‘array’ dan memiliki tipe data ‘i’ dan elemen [1,2,3,4,5].
- Baris selanjutnya akan mencetak string "Array before updation : " dan menggunakan loop ‘for’, program ini mencetak setiap elemen array ‘arr’.
- ‘/r’ berfungsi untuk mencetak baris baru.
- Selanjutnya akan mengubah elemen ‘arr’ pada indeks ke-2 menjadi nilai 6.
- Kemudian akan mencetak string "Array after updation:" menggunakan loop ‘for’ dan mencetak setiap elemen array ‘arr’ yang telah diubah.
- Dan baris selanjutnya akan mengubah elemen ‘arr’ pada indeks ke-4 menjadi nilai 8.
- Terakhir akan mencetak string "Array after updation : " kemudian menggunakan loop ‘for’ dan mencetak setiap elemen array ‘arr’ setelah diubah.

7. Operasi pada Array Python

Untuk menghitung elemen di dalam sebuah array, diperlukan metode count.

```
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
```

(Gambar 2.7)

- Mengimport modul ‘array’

- Baris selanjutnya membuat array dengan nama ‘my_array’ menggunakan fungsi ‘arary()’ dari modul ‘array’ yang memiliki tipe data ‘i’ dan elemen [1,2,3,4,5,2]
- Selanjutnya menggunakan metode ‘count()’ yang berfungsi untuk mengitung jumlah kemunculan nilai 2 dalam array dan menetapkan variabel ‘count’.
- Baris terakhir mencetak string "Number of occurrences of 2:" diikuti oleh variabel ‘vount’.

8. Membalikkan Elemen dalam Array

Untuk membalikkan elemen-elemen yang ada di dalam sebuah array, diperlukan 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
```

(Gambar 2.8)

- Mengimport modul ‘array’
- Kemudian membuat array dengan nama ‘my_array’ menggunakan fungsi ‘array()’ dari modul ‘array’ dan memiliki tipe data ‘i’ dengan elemen [1,2,3,4,5,6,2]
- Lalu menggunakan metode ‘count()’ untuk menghitung jumlah kemunculan 2 di array dan menetapkan variabel ‘count’
- Program akan mencetak string "Original array:" diikuti oleh isi array ‘my_array’ menggunakan operator ‘**’ yang mengonversikan menjadi argumen terpisah.
- Kemudian memakai metode ‘reverse()’ untuk membalik urutan elemen dalam array.
- Baris terakhir mencetak string "Reversed array:" diikuti oleh isi array ‘my_array’ yangg sudah dibalik menggunakan operator ‘*’ dan menjadikan argumen terpisah.

9. Extending Elemen dalam Array

Dalam Python, array berfungsi untuk penyimpanan beberapa nilai atau nilai dengan tipe yang sama dalam suatu variabel. Fungsi extend() bisa dipakai untuk menambahkan nilai item dari iterable ke bagian akhir array.

```
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
```

(Gambar 2.9)

- Import modul ‘array’ dan berikan nama alias ‘arr’.
- Membuat array dengan nama ‘a’ menggunakan fungsi ‘array()’ yang memiliki tipe data ‘i’ dan elemen [1,2,3,4,5].
- Kemudian mencetak string "The before array extend : " dan menggunakan ‘for’, juga mencetak setiap elemen array ‘a’.
- Selanjutnya menggunakan metode ‘extend()’ untuk menambahkan beberapa elemen baru ke dalam array ‘a’.
- Terakhir mencetak string "The array after extend : " dan menggunakan loop ‘for’ lalu akan mencetak elemen array ‘a’.

```
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
```

(Gambar 2.10)

- Import modul ‘array’
- Kemudian membuat array dengan nama ‘a’ menggunakan fungsi ‘array()’ dengan tipe data ‘i’ dan elemen [1,2,3,4,5,6].
- Selanjutnya program akan mencetak string "The Before extend array is " lalu menggunakan loop ‘for’, baris ini mencetak setiap elemen array ‘a’.
- Baris selanjutnya memakai metode ‘extend()’ untuk menambahkan beberapa elemen baru ke array ‘a’
- Lalu program akan mencetak string "The After extend array is ", menggunakan loop ‘for’ dengan mencetak setiap elemen array ‘a’ setelah dilakukan penambahan.
- Pada baris selanjutnya membuat array dengan nama ‘b’ memakai fungsi ‘array()’ dan memiliki tipe data ‘d’ dan elemen [2.1,2.2,2.3,2.4,2.5,2.6].
- Selanjutnya program mencetak string "The before extend array is” lalu menggunakan loop ‘for’ dan mencetak setiap elemen array ‘b’.
- Kemudian menggunakan metode ‘estend()’ untuk menambahkan beberapa elemen baru ke array ‘b’
- Pada baris terakhir akan mencetak string "The after extend array is " lalu menggunakan loop ‘for’ dan mencetak setiap elemen array ‘b’ setelah penambahan

C. Kesimpulan

Dalam penggunaan array di Python, kita dapat melakukan berbagai operasi untuk mengelola data. Pertama, kita dapat membuat array dengan menentukan tipe data elemen dan memberikan nilai awalnya. Setelah membuatnya, kita dapat menambahkan elemen baru menggunakan metode insert() untuk menyisipkan elemen di posisi tertentu atau append() untuk menambahkan elemen di akhir array. Untuk mengakses elemen dalam array, kita hanya perlu menggunakan indeksnya. Jika kita perlu menghapus elemen dari array, kita bisa menggunakan metode pop() untuk menghapus berdasarkan indeks atau remove() untuk menghapus berdasarkan nilai.

Selain itu, kita juga bisa melakukan operasi seperti menghitung jumlah kemunculan suatu nilai dalam array. Jika kita perlu memanipulasi sebagian data dalam array, kita dapat menggunakan teknik pemotongan array dengan slicing. Untuk mengubah nilai elemen dalam array, kita hanya perlu menetapkan nilai baru ke indeks yang diinginkan. Ada juga opsi untuk membalik urutan elemen dalam array menggunakan metode reverse(). Akhirnya, jika kita perlu menambahkan beberapa elemen baru ke dalam array yang sudah ada, kita bisa menggunakan metode extend(). Dengan pemahaman yang baik tentang operasi-operasi ini, kita dapat mengelola array secara efisien dalam pengembangan aplikasi Python.

D. Referensi

Fahrudin, T. M., & S ST, M. T. 2023. *Algoritma dan Pemrograman Dasar dalam Bahasa Pemrograman Python*. Thalibul Ilmi Publishing & Education.

Kurniawan, D. 2022. *Pengenalan Machine Learning dengan Python*. Elex Media Komputindo.