Nama : Nur Taufiq Hidayat

NIM : L200180069

Kelas : C

MODUL 3

Collections, Arrays, and Linked Structures

Latihan 3.1 Membuat matrix dengan ordo 2 x 2.

```
Latihan 3.1 Membuat Matrix 2 x 2
>>> A = [ [2,3], [5,7] ]
>>> A [0][1]
3
>>> A[1][1]
7
>>> A[0][0]
2
>>> A[1][0]
5
>>>
```

Latihan 3.2 Membuat matrix 3 x 3 berisi 0 semua.

```
Latihan 3.2 Membuat Matrix 3 x 3 berisi 0
>>> B = [ [0 for j in range(3)] for i in range(3)]
>>> B
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
>>>
```

List Comprehension

- Membuat list kuadrat bilangan dari 0 sampai 6
  ```
  >>> [x**2 for x in range(0, 7)]
  [0, 1, 4, 9, 16, 25, 36]
  ```
- Membuat list yang berisi tuple pasangan bilangan dan kuadratnya, dari 0 sampai 6
  ```
  >>> [(x, x**2) for x in range(7)]
  [(0, 0), (1, 1), (2, 4), (3, 9), (4, 16), (5, 25), (6, 36)]
  ```
- Membuat list kuadrat bilangan genap antara 0 sampai 15
  ```
  >>> [x**2 for x in range(15) if x%2 == 0]
  [0, 4, 16, 36, 64, 100, 144, 196]
  ```
- Membuat list sepanjang 5 elemen yang berisi bilangan 3
  ```
  >>> [3 for i in range(5)]
  [3, 3, 3, 3, 3]
  ```

- Membuat list sepanjang 3 elemen yang berisi list sepanjang 3 elemen angka 0

```
>>> [[0 for j in range(3)]for i in range (3)]
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
```

- Membuat matrix identitas 3 x 3

```
>>> [[1 if j==i else 0 for j in range (3)] for i in range (3)]
[[1, 0, 0], [0, 1, 0], [0, 0, 1]]
```

- Membuat list yang berisi huruf vocal suatu string

```
>>> d = 'Yogyakarta dan Surakarta'
>>> [x for x in d if x in 'aiueoAIUEO']
['o', 'a', 'a', 'a', 'a', 'u', 'a', 'a', 'a']
```

- Membuat list bilangan genap dari 20 sampai 50

```
>>> [x for x in range(20,50) if apakahGenap(x)]
[20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48]
>>>
```

Linked Structures

- Linked List

```
class Node(object) :
    '''Sebuah simpul di linked list'''
    def __init__(self, data, next=None):
        self.data = data
        self.next = next
```

Saat dijalankan di Python Shell :

```
>>> a = Node(11)
>>> b = Node(52)
>>> c = Node(18)
>>> d = Node(7)
>>> e = Node(31)
>>> a.next = b
>>> b.next = c
>>> c.next = d
>>> d.next = e
>>> print(a.data)
11
>>> print(a.next.data)
52
>>> print(a.next.next.data)
18
>>> print(c.next.data)
7
>>> print(b.next.next.next.data)
31
```

- Mengunjungi setiap simpul dari depan

```
class Node(object) :
    '''Sebuah simpul di linked list'''
    def __init__(self, data, next=None):
        self.data = data
        self.next = next
    def kunjungi(self, head):
        curNode = head
        while curNode is not None:
            print(curNode.data)
            curNode = curNode.next
```

Saat dijalankan di Python Shell :

```
>>> a = Node(11)
>>> b = Node(52)
>>> c = Node(18)
>>> d = Node(7)
>>> e = Node(31)
>>> a.next = b
>>> b.next = c
>>> c.next = d
>>> d.next = e
>>> a.kunjungi(a)
11
52
18
7
31
>>> d.kunjungi(d)
7
31
```

- Advanced Linked List

```
class DNode(object):
    def __init__(self, data):
        self.data = data
        self.next = None
        self.prev = None
```

Saat dijalankan di Python Shell :

```
>>> a = DNode(11)
>>> b = DNode(52)
>>> c = DNode(18)
>>> d = DNode(7)
>>> e = DNode(31)
>>> a.next = b
>>> b.next = c
>>> c.next = d
>>> e.prev = d
>>> c.prev = b

>>> print(a.next.data)
52
>>> print(e.prev.data)
7
```