Nama : Salmaa Khoirun Nisaa’

NIM : L200180019

Kelas : A

Modul 3

Collections, Array, and Linked

1. **Tugas 1**

```
def cekMatrix(matrix):
    hitung = 0
    if len(matrix) != len(matrix[0]):
        print('Matriks haruslah bujur sangkar')
        return
    for i in range(len(matrix)):
        cek = all(isinstance(x, int) for x in matrix[i])
        if cek == True:
            hitung += 1
    if hitung == len(matrix):
        print('Matriks bujur sangkar dan memiliki tipe yang konsisten')
    else:
        print('Matriks memiliki tipe yang tidak konsisten')


def ukurMatrix(matrix):
    ukurBaris = len(matrix)
    ukurKolom = len(matrix[0])
```

```
    print('Matriks tersebut adalah matriks berukuran {}x{}'.format(ukurBaris, ukurKolom))

def penjumlahanMatrix(matrix1, matrix2):
    if len(matrix1) != len(matrix2):
        print('Ukuran matriks tidak sama')
        return

    jumlah = [[0 for j in range(len(matrix1[0]))] for i in range(len(matrix1))]

    for i in range(len(matrix1)):
        for j in range(len(matrix2)):
            jumlah[i][j] = matrix1[i][j] + matrix2[i][j]

    for i in jumlah:
        print(i)

def perkalianMatrix(matrix1, matrix2):
    if len(matrix1) != len(matrix2):
        print('Ukuran matriks tidak sesuai')
        return

    kali = [[0 for j in range(len(matrix1[0]))] for i in range(len(matrix1))]

    for i in range(len(matrix1)):
        for j in range(len(matrix1)):
```

```
            for k in range(len(matrix1)):
                kali[i][j] += matrix1[i][k] * matrix2[k][j]

    for i in kali:
        print(i)

def determinanMatrix(matrix):
    if len(matrix) != len(matrix[0]):
        print('Matriks harus bujur sangkar')
        return

    tambah = [1 for i in range(len(matrix))]
    kurang = [1 for i in range(len(matrix))]

    for i in range(len(matrix)):
        for j in range(len(matrix) - 1):
            matrix[i].append(matrix[i][j])

    matrix2 = matrix.copy()

    for i in range(len(matrix2)):
        matrix2[i] = list(reversed(matrix2[i]))

    nilai = 0
    for i in range(len(matrix)):
```

```
    for j in range(len(matrix)):
        tambah[i] *= matrix[j][j + nilai]
        kurang[i] *= matrix2[j][j + nilai]
    nilai += 1

kurang = [-x for x in kurang]
determinan = sum(tambah) + sum(kurang)

return determinan
```

```
>>> cekMatrix(m1)
Matriks bujur sangkar dan memiliki tipe yang konsisten
>>> m2 = [[1,2,3],[1,3,4],[1,4,3]]
>>> cekMatrix(m2)
Matriks bujur sangkar dan memiliki tipe yang konsisten
>>> m3 = [[1,2,3],[1,2,5]]
>>> cekMatrix(m3)
Matriks haruslah bujur sangkar
>>> m4 = [[3,4,5],[1,2,3],[2,1,'5']]
>>> cekMatrix(m4)
Matriks memiliki tipe yang tidak konsisten
>>> ukurMatrix(m1)
Matriks tersebut adalah matriks berukuran 3x3
>>> penjumlahanMatrix(m1,m2)
[2, 4, 6]
[5, 8, 10]
[8, 12, 12]
>>> perkalianMatrix(m1,m2)
[6, 20, 20]
[15, 47, 50]
[24, 74, 80]
>>> determinanMatrix(m1)
0
>>> determinanMatrix(m2)
-2
>>>
```

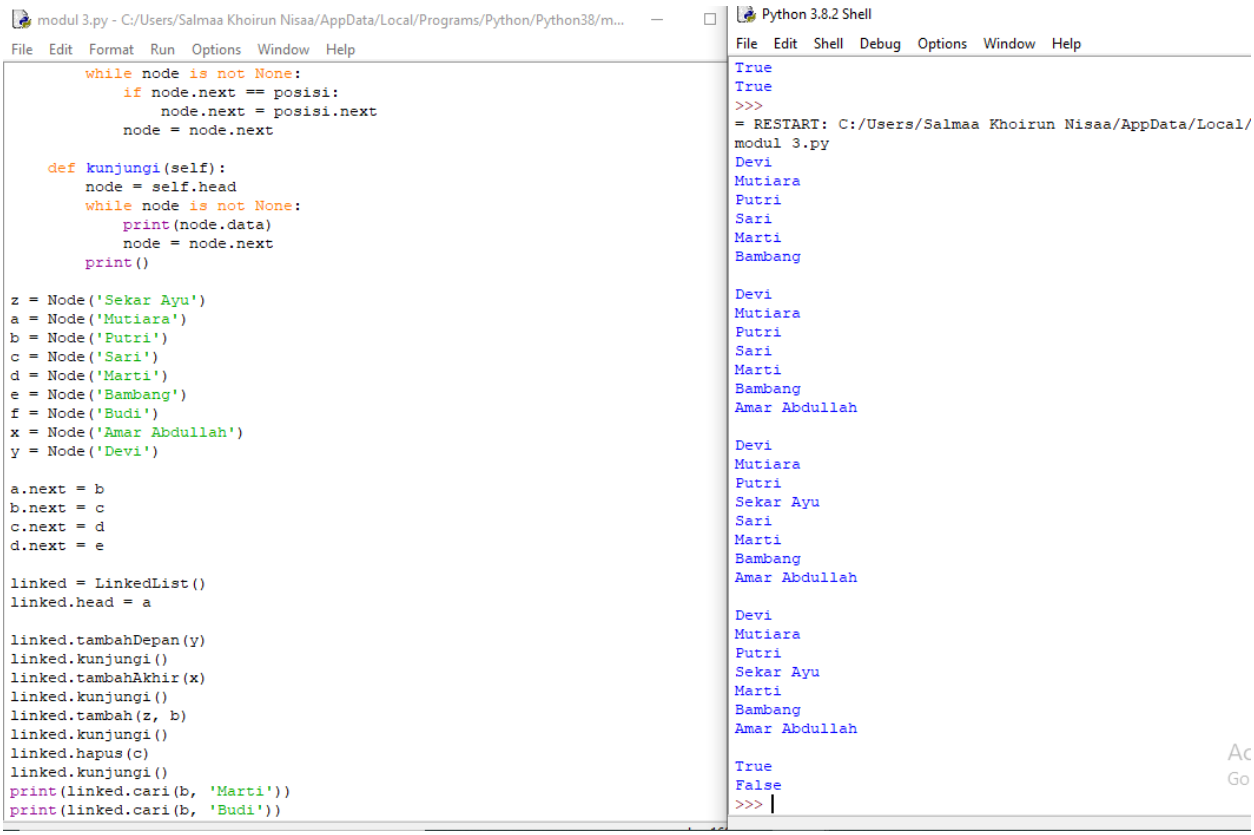
2. Tugas 2

```
#Matrix list comprehension
def buatNol(m, n = None):
    if n == None:
        n = m

    matrix = [[0 for j in range(n)] for i in range(m)]

    for i in matrix:
        print(i)

def buatIdentitas(m):
    matrix = [[ 1 if j == i else 0 for j in range(m)] for i in range(m)]

    for i in matrix:
        print(i)
```

```
= RESTART: C:/Users/Salmaa Khoirun Nisaa/AppData/Local/Programs/Pytho
modul 3.py
>>> buatNol(3)
[0, 0, 0]
[0, 0, 0]
[0, 0, 0]
>>> buatIdentitas(6)
[1, 0, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 0]
[0, 0, 1, 0, 0, 0]
[0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 0, 1]
>>>
```


3. Tugas 3

```python
#linked list
class Node:
    def __init__(self, data, next = None):
        self.data = data
        self.next = next

class LinkedList:
    def __init__(self, head = None):
        self.head = head

    def cari(self, head, yang_dicari):
        while head is not None:
            if head.data == yang_dicari:
                return True
            head = head.next
        return False

    def tambahDepan(self, head):
        head.next = self.head
        self.head = head

    def tambahAkhir(self, head):
        node = self.head
        while node.next != None:
            node = node.next
        node.next = head

    def tambah(self, head, posisi):
        head.next = posisi.next
        posisi.next = head

    def hapus(self, posisi):
        node = self.head
        while node is not None:
            if node.next == posisi:
                node.next = posisi.next
            node = node.next

    def kunjungi(self):
        node = self.head
```


modul 3.py - C:/Users/Salmaa Khoirun Nisaa/AppData/Local/Programs/Python/Python38/m...
File Edit Format Run Options Window Help
while node is not None:
if node.next == posisi:
node.next = posisi.next
node = node.next
def kunjungi(self):
node = self.head
while node is not None:
print(node.data)
node = node.next
print()
z = Node('Sekar Ayu')
a = Node('Mutiara')
b = Node('Putri')
c = Node('Sari')
d = Node('Marti')
e = Node('Bambang')
f = Node('Budi')
x = Node('Amar Abdullah')
y = Node('Devi')
a.next = b
b.next = c
c.next = d
d.next = e
linked = LinkedList()
linked.head = a
linked.tambahDepan(y)
linked.kunjungi()
linked.tambahAkhir(x)
linked.kunjungi()
linked.tambah(z, b)
linked.kunjungi()
linked.hapus(c)
linked.kunjungi()
print(linked.cari(b, 'Marti'))
print(linked.cari(b, 'Budi'))
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
True
True
>>>
= RESTART: C:/Users/Salmaa Khoirun Nisaa/AppData/Local/
modul 3.py
Devi
Mutiara
Putri
Sari
Marti
Bambang
Devi
Mutiara
Putri
Sari
Marti
Bambang
Amar Abdullah
Devi
Mutiara
Putri
Sekar Ayu
Sari
Marti
Bambang
Amar Abdullah
Devi
Mutiara
Putri
Sekar Ayu
Marti
Bambang
Amar Abdullah
True
False
>>>

4. Tugas 4

```python
#link doubly list

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

class DoublyLinkedList:
    def __init__(self, head = None):
        self.head = head

    def kunjungi(self):
        node = self.head
        while node is not None:
            print(node.data)
            reverse = node
            node = node.next
        print()
        while reverse is not None:
            print(reverse.data)
            reverse = reverse.prev
        print()

    def tambahAwal(self, head):
        head.next = self.head
        self.head = head
        head.next.prev = head
```

```python
    def tambahAwal(self, head):
        head.next = self.head
        self.head = head
        head.next.prev = head

z = DNode('Sekar Ayu')
a = DNode('Mutiara')
b = DNode('Putri')
c = DNode('Sari')
d = DNode('Marti')
e = DNode('Bambang')
f = DNode('Budi')
x = DNode('Amar Abdullah')
y = DNode('Devi')

a.next = b
b.prev = a
b.next = c
c.prev = b
c.next = d
d.prev = c
d.next = e
e.prev = d
e.next = f
f.prev = e

doubly = DoublyLinkedList()
doubly.head = a

doubly.kunjungi()
doubly.tambahAwal(z)
doubly.kunjungi()
```

```
>>>
= RESTART: C:/Users/Salmaa Khoirun Nisaa/Appl
modul 3.py
Mutiara
Putri
Sari
Marti
Bambang
Budi

Budi
Bambang
Marti
Sari
Putri
Mutiara

Sekar Ayu
Mutiara
Putri
Sari
Marti
Bambang
Budi

Budi
Bambang
Marti
Sari
Putri
Mutiara
Sekar Ayu

>>>
```