Nama: Ghani Prasetia

NIM : L200180185

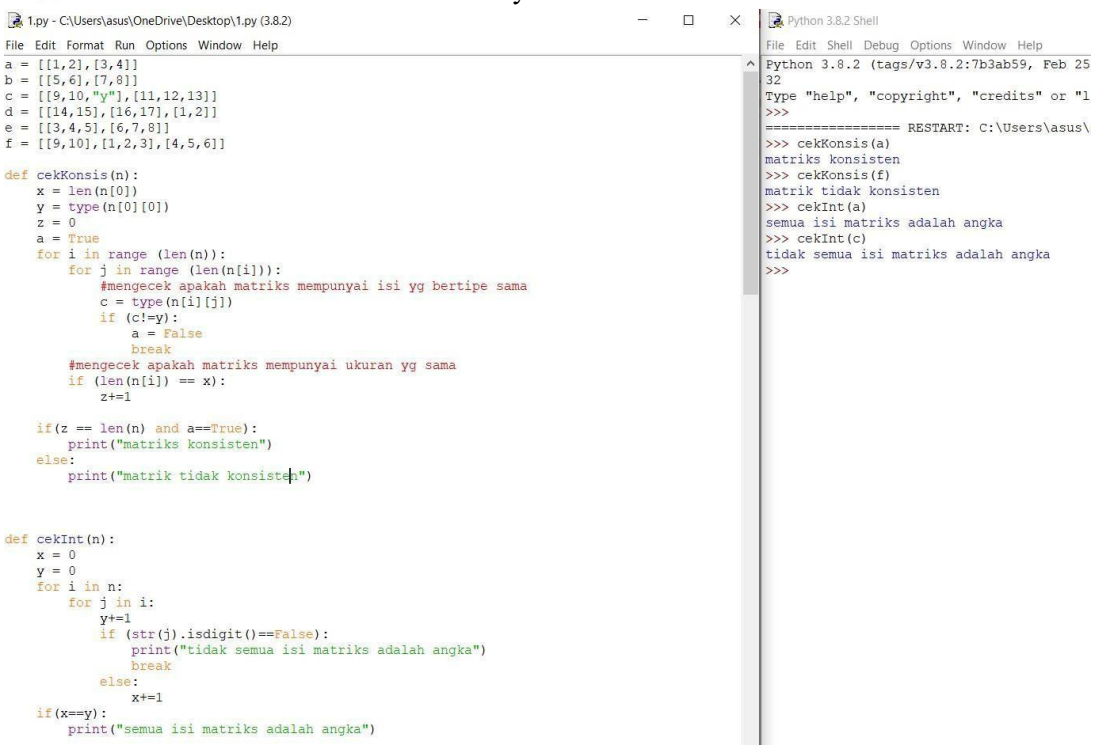
Kelas : G

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

Modul 3

Terkait array dua dimensi,Membuat tipe data sebuah matriks yang berisi angka angka

- Memastikan isi dan ukuran matrix-nya konsisten



1.py - C:\Users\asus\OneDrive\Desktop\1.py (3.8.2)

File Edit Format Run Options Window Help

```
a = [[1,2],[3,4]]
b = [[5,6],[7,8]]
c = [[9,10,"y"],[11,12,13]]
d = [[14,15],[16,17],[1,2]]
e = [[3,4,5],[6,7,8]]
f = [[9,10],[1,2,3],[4,5,6]]

def cekKonsis(n):
    x = len(n[0])
    y = type(n[0][0])
    z = 0
    a = True
    for i in range (len(n)):
        for j in range (len(n[i])):
            #mengecek apakah matriks mempunyai isi yg bertipe sama
            c = type(n[i][j])
            if (c!=y):
                a = False
                break
        #mengecek apakah matriks mempunyai ukuran yg sama
        if (len(n[i]) == x):
            z+=1

    if(z == len(n) and a==True):
        print("matriks konsisten")
    else:
        print("matrik tidak konsisten")


def cekInt(n):
    x = 0
    y = 0
    for i in n:
        for j in i:
            y+=1
            if (str(j).isdigit()==False):
                print("tidak semua isi matriks adalah angka")
                break
            else:
                x+=1
    if(x==y):
        print("semua isi matriks adalah angka")
```

Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25
32
Type "help", "copyright", "credits" or "l
>>>
================ RESTART: C:\Users\asus\
>>> cekKonsis(a)
matriks konsisten
>>> cekKonsis(f)
matrik tidak konsisten
>>> cekInt(a)
semua isi matriks adalah angka
>>> cekInt(c)
tidak semua isi matriks adalah angka
>>>
```

Mengambil ukuran matriks nya

1.py - C:\Users\asus\OneDrive\Desktop\1.py (3.8.2)

File Edit Format Run Options Window Help

```
d = [[14,15],[16,17],[1,2]]
e = [[3,4,5],[6,7,8]]
f = [[9,10],[1,2,3],[4,5,6]]

def ordo(n):
    x,y = 0,0
    for i in range(len(n)):
        x+=1
        y = len(n[i])
    print(len(n))
    print("mempunyai ordo "+str(x)+"x"+str(y))
```

Python 3.8.2 Shell

File Edit Shell Debug Options

```
Python 3.8.2 (tags/v3.8.2:
tel)] on win32
Type "help", "copyright",
>>>
================== RESTART:
>>> ordo(a)
2
mempunyai ordo 2x2
>>> ordo(e)
2
mempunyai ordo 2x3
>>>
```

Menjumlahkan dua matriks

1.py - C:\Users\asus\OneDrive\Desktop\1.py (3.8.2)

File Edit Format Run Options Window Help

```
a = [[1,2],[3,4]]
b = [[5,6],[7,8]]
c = [[9,10,"y"],[11,12,13]]
d = [[14,15],[16,17],[1,2]]
e = [[3,4,5],[6,7,8]]
f = [[9,10],[1,2,3],[4,5,6]]

def jumlah(n,m):
    x,y = 0,0
    for i in range(len(n)):
        x+=1
        y = len(n[i])
    xy = [[0 for j in range(x)] for i in range(y)]

    z = 0
    if(len(n)==len(m)):
        for i in range(len(n)):
            if(len(n[i]) == len(m[i])):
                z+=1
    if(z==len(n) and z==len(m)):
        print("ukuran sama")
        for i in range(len(n)):
            for j in range(len(n[i])):
                xy[i][j] = n[i][j] + m[i][j]
        print(xy)
    else:
        print("ukuran beda")
```

Python 3.8.2 Shell

File Edit Shell Debug Options Windc

```
Python 3.8.2 (tags/v3.8.2:7b3
tel)] on win32
Type "help", "copyright", "cr
>>>
================== RESTART: C:
>>> jumlah(a,b)
ukuran sama
[[6, 8], [10, 12]]
>>> jumlah(a,d)
ukuran beda
>>>
```

Mengalikan dua matriks

1.py - C:\Users\asus\OneDrive\Desktop\1.py (3.8.2)

File Edit Format Run Options Window Help

```
a = [[1,2],[3,4]]
b = [[5,6],[7,8]]
c = [[9,10,"y"],[11,12,13]]
d = [[14,15],[16,17],[1,2]]
e = [[3,4,5],[6,7,8]]
f = [[9,10],[1,2,3],[4,5,6]]


def kali(n,m):
    aa = 0
    x,y = 0,0
    for i in range(len(n)):
        x+=1
        y = len(n[i])
    v,w = 0,0
    for i in range(len(m)):
        v+=1
        w = len(m[i])

    if(y==v):
        print("bisa dikalikan")
        vwxy = [[0 for j in range(w)] for i in range(x)]
        print(vwxy)
        for i in range(len(n)):
            for j in range(len(m[0])):
                for k in range(len(m)):
                    #print(n[i][k], m[k][j])
                    vwxy[i][j] += n[i][k] * m[k][j]
        print(vwxy)

    else:
        print("tidak memenuhi syarat")
```

Python 3.8.2 Shell

File Edit Shell Debug Options Window

```
Python 3.8.2 (tags/v3.8.2:7b3a
tel)] on win32
Type "help", "copyright", "cre
>>>
================ RESTART: C:\
>>> kali(a,e)
bisa dikalikan
[[0, 0, 0], [0, 0, 0]]
[[15, 18, 21], [33, 40, 47]]
>>> zz = [[1,2,3],[1,2,3]]
>>> zx = [[1],[2],[3]]
>>> kali(zz,zx)
bisa dikalikan
[[0], [0]]
[[14], [14]]
>>>
```

Menghitung determinan sebuah matriks

1.py - C:\Users\asus\OneDrive\Desktop\1.py (3.8.2)

File Edit Format Run Options Window Help

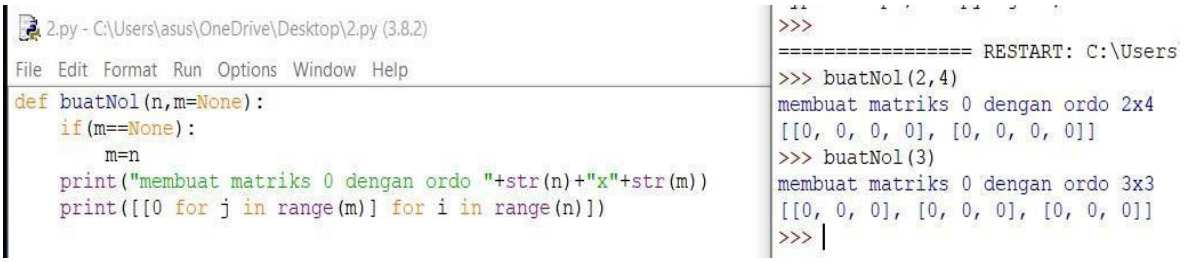
```
a = [[1,2],[3,4]]
b = [[5,6],[7,8]]
c = [[9,10,"y"],[11,12,13]]
d = [[14,15],[16,17],[1,2]]
e = [[3,4,5],[6,7,8]]
f = [[9,10],[1,2,3],[4,5,6]]

def determHitung(A, total=0):
    x = len(A[0])
    z = 0
    for i in range(len(A)):
        if (len(A[i]) == x):
            z+=1
    if(z == len(A)):
        if(x==len(A)):
            indices = list(range(len(A)))
            if len(A) == 2 and len(A[0]) == 2:
                val = A[0][0] * A[1][1] - A[1][0] * A[0][1]
                return val
            for fc in indices:
                As = A
                As = As[1:]
                height = len(As)
                for i in range(height):
                    As[i] = As[i][0:fc] + As[i][fc+1:]
                sign = (-1) ** (fc % 2)
                sub_det = determHitung(As)
                total += sign * A[0][fc] * sub_det
        else:
            return "tidak bisa dihitung determinan, bukan matrix bujursangkar"
    else:
        return "tidak bisa dihitung determinan, bukan matrix bujursangkar"
    return total
```

Python 3.8.2 Shell

File Edit Shell Debug Options Wi

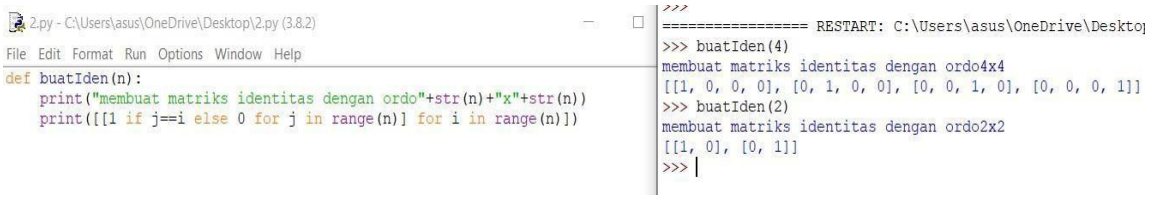
```
Python 3.8.2 (tags/v3.8.2:7
tel)] on win32
Type "help", "copyright", "
>>>
================ RESTART:
>>> z = [[3,1],[2,5]]
>>> determHitung(z)
13
>>> r = [[10,23,45,12,13],
     [1,2,3,4,5],
     [1,2,3,4,6],
     [4,2,3,4,8],
     [1,4,5,6,10]]
>>> determHitung(r)
330
>>>
```

Terkait matriks dan list comprehension, membuat fungsi memanfaatkan list comprehension
Untuk membangkitkan matriks 0 semua



.

Untuk membangkitkan matriks identitas



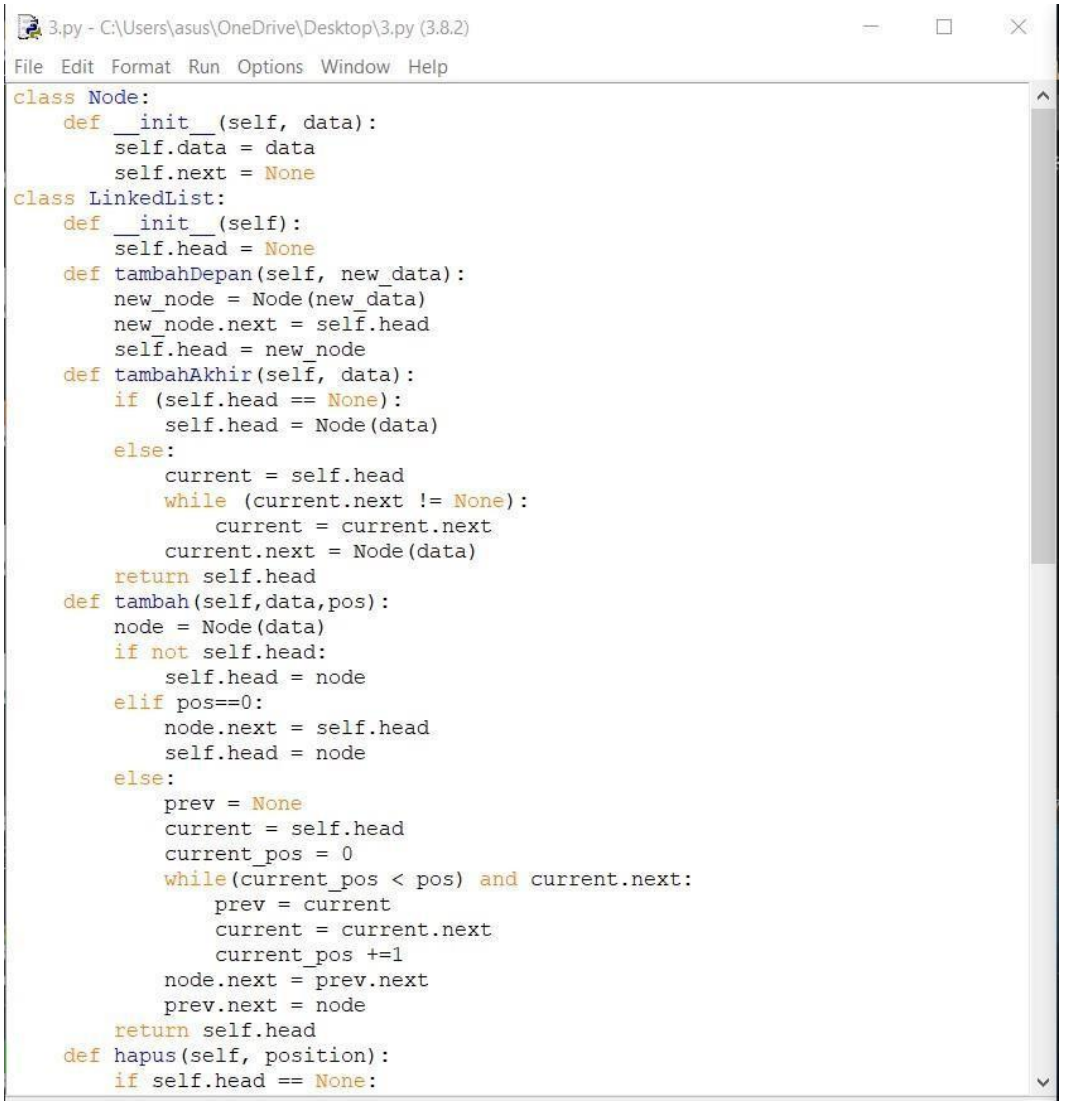
Terkait linked list, membuat fungsi

Mencari data yang isinya tertentu : cari(head,yang_dicari)

Menambah suatu simpul di awal : tambahDepan(head)

Menambah suatu simpul di akhir : tambahAkhir(head)

Menyisipkan suatu simpul di mana saja : tambah(head,posisi)
Menghapus suatu simpul di awal, diakhir, dan dimana saja : hapus (posisi)




```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def tambahDepan(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node
    def tambahAkhir(self, data):
        if (self.head == None):
            self.head = Node(data)
        else:
            current = self.head
            while (current.next != None):
                current = current.next
            current.next = Node(data)
        return self.head
    def tambah(self,data,pos):
        node = Node(data)
        if not self.head:
            self.head = node
        elif pos==0:
            node.next = self.head
            self.head = node
        else:
            prev = None
            current = self.head
            current_pos = 0
            while(current_pos < pos) and current.next:
                prev = current
                current = current.next
                current_pos +=1
            node.next = prev.next
            prev.next = node
        return self.head
    def hapus(self, position):
        if self.head == None:
```

3.py - C:\Users\asus\OneDrive\Desktop\3.py (3.8.2)

File Edit Format Run Options Window Help

```
        if self.head == None:
            return
        temp = self.head
        if position == 0:
            self.head = temp.next
            temp = None
            return
        for i in range(position ):
            prev = temp
            temp = temp.next
            if temp is None:
                break
        if temp is None:
            return
        if temp.next is None:
            return
        prev.next = temp.next
        temp= None

    def cari(self, x):
        current = self.head
        while current != None:
            if current.data == x:
                return "True"
            current = current.next
        return "False"
    def display(self):
        current = self.head
        while current is not None:
            print(current.data, end = ' ')
            current = current.next

llist = LinkedList()
llist.tambahDepan(21)
llist.tambahDepan(22)
llist.tambahDepan(12)
llist.tambahDepan(14)
llist.tambahDepan(2)
llist.tambahDepan(19)
llist.tambahAkhir(9)
llist.display()
llist.hapus(5)
llist.tambah(1,5)
print(llist.cari(21))
print(llist.cari(29))
llist.display()
```

Python 3.8.2 Shell

File Edit Shell Debug Options Windo

```
Python 3.8.2 (tags/v3.8.2:7b3
tel)] on win32
Type "help", "copyright", "cr
>>>
================== RESTART: C:
19 2 14 12 22 21 9 False
False
19 2 14 12 22 1 9
>>>
```

Terkait doubly linked lis, membuat fungsi
Mengunjungi dan mencetak data tiap simpul dari depan dan dari belakang Menambah suatu simpul di awal

Menambah suatu simpul di akhir

4.py - C:\Users\asus\OneDrive\Desktop\4.py (3.8.2)

File Edit Format Run Options Window Help

```
class Node:
    def __init__(self, data):
        self.data = data
        self.prev = None
class DoublyLinkedList:
    def __init__(self):
        self.head = None
    def awal(self, new_data):
        print("menambah pada awal", new_data)
        new_node = Node(new_data)
        new_node.next = self.head
        if self.head is not None:
            self.head.prev = new_node
        self.head = new_node
    def akhir(self, new_data):
        print("menambah pada akhir", new_data)
        new_node = Node(new_data)
        new_node.next = None
        if self.head is None:
            new_node.prev = None
            self.head = new_node
            return
        last = self.head
        while(last.next is not None):
            last = last.next
        last.next = new_node
        new_node.prev = last
        return
    def printList(self, node):
        print("\nDari Depan :")
        while(node is not None):
            print(" % d" %(node.data))
            last = node
            node = node.next
        print("\nDari Belakang :")
        while(last is not None):
            print(" % d" %(last.data))
            last = last.prev
llist = DoublyLinkedList()
llist.awal(7)
llist.awal(1)
llist.akhir(6)
llist.akhir(4)
llist.printList(llist.head)
```

Python 3.8.2 Shell

File Edit Shell Debug Options

```
Python 3.8.2 (tags/v3.8.
tel)] on win32
Type "help", "copyright"
>>>
================ RESTAR
menambah pada awal 7
menambah pada awal 1
menambah pada akhir 6
menambah pada akhir 4

Dari Depan :
  1
  7
  6
  4

Dari Belakang :
  4
  6
  7
  1
>>>
```