

Nama : Fatwa Dwi Hidayat

NIM : L200180176

Kelas : G

## LAPORAN PRAKTIKUM MODUL 3

### ALGORITMA DAN STRUKTUR DATA

1. 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 V

```
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 Win
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
>>> |
```

## 2. Terkait matriks dan list comprehension, membuat fungsi memanfaatkan list comprehension

- Untuk membangkitkan matriks 0 semua

```
2.py - C:\Users\asus\OneDrive\Desktop\2.py (3.8.2)
File Edit Format Run Options Window Help
def buatNol(n,m=None):
    if(m==None):
        m=n
    print("membuat matriks 0 dengan ordo "+str(n)+"x"+str(m))
    print([[0 for j in range(m)] for i in range(n)])

>>>
===== RESTART: C:\Users\
>>> buatNol(2,4)
membuat matriks 0 dengan ordo 2x4
[[0, 0, 0, 0], [0, 0, 0, 0]]
>>> buatNol(3)
membuat matriks 0 dengan ordo 3x3
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
>>> |
```

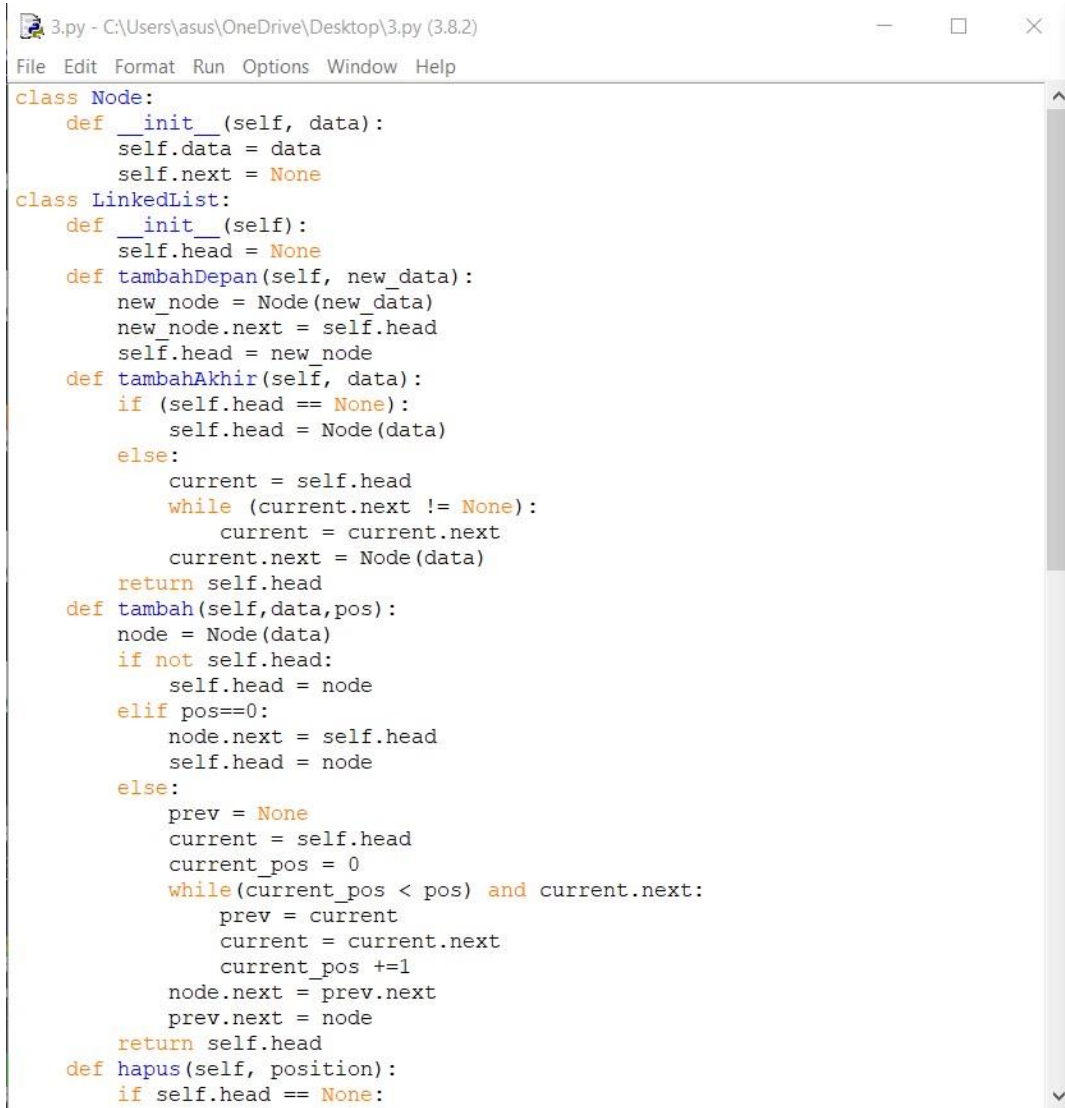
- Untuk membangkitkan matriks identitas

```
2.py - C:\Users\asus\OneDrive\Desktop\2.py (3.8.2)
File Edit Format Run Options Window Help
def buatIden(n):
    print("membuat matriks identitas dengan ordo"+str(n)+"x"+str(n))
    print([[1 if j==i else 0 for j in range(n)] for i in range(n)])

>>>
===== RESTART: C:\Users\asus\OneDrive\Desktop
>>> buatIden(4)
membuat matriks identitas dengan ordo4x4
[[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
>>> buatIden(2)
membuat matriks identitas dengan ordo2x2
[[1, 0], [0, 1]]
>>> |
```

### 3. 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)

A screenshot of a Python IDE window titled '3.py - C:\Users\asus\OneDrive\Desktop\3.py (3.8.2)'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code is written in Python and implements a linked list with two classes: 'Node' and 'LinkedList'. The 'Node' class has an '\_\_init\_\_' method that takes 'self' and 'data' as arguments, sets 'self.data' to 'data', and initializes 'self.next' to 'None'. The 'LinkedList' class has an '\_\_init\_\_' method that sets 'self.head' to 'None'. It also has four methods: 'tambahDepan' (adds a new node at the beginning), 'tambahAkhir' (adds a new node at the end), 'tambah' (inserts a new node at a specific position), and 'hapus' (removes a node at a specific position). The code is color-coded with syntax highlighting.

```
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

l1ist = LinkedList()
l1ist.tambahDepan(21)
l1ist.tambahDepan(22)
l1ist.tambahDepan(12)
l1ist.tambahDepan(14)
l1ist.tambahDepan(2)
l1ist.tambahDepan(19)
l1ist.tambahAkh1r(9)
l1ist.display()
l1ist.hapus(5)
l1ist.tambah(1,5)
print(l1ist.cari(21))
print(l1ist.cari(29))
l1ist.display()

```

```

Python 3.8.2 Shell
File Edit Shell Debug Options Window
Python 3.8.2 (tags/v3.8.2:7b3e36e, Feb 20 2020) on win32
Type "help", "copyright", "credits()" or "quit()" for more
>>>
===== RESTART: C:\Python38\Python.exe
19 2 14 12 22 21 9 False
False
19 2 14 12 22 1 9
>>> |

```

#### 4. Terkait doubly linked list, 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
l1ist = DoublyLinkedList()
l1ist.awal(7)
l1ist.awal(1)
l1ist.akhir(6)
l1ist.akhir(4)
l1ist.printList(l1ist.head)

Python 3.8.2 Shell
File Edit Shell Debug Options
Python 3.8.2 (tags/v3.8.
tel)] on win32
Type "help", "copyright"
>>>
===== RESTART
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
>>> |