LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
MODUL 3
COLLECTIONS, ARRAYS, AND LINKED STRUCTURES

Disusun oleh:
Bima Triadmaja
L200210137
E

TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2022/2023

3.4 Soal-soal untuk Mahasiswa

Nomor 1 :

```python
#NO.1
a = [[1,2],[3,4]]
b = [[5,6],[7,8]]
c = [[12,3,"x","y"],[12,33,4]]
d = [[3,4],[2,4],[1,5]]
e = [[5,6,7],[7,8,9]]
f = [[1,2,3],[4,5,6],[7,8,9]]

def cekConsisten(n):
    x = len(n[0])
    z = 0
    for i in range(len(n)):
        if (len(n[i]) == x):
            z+=1
    if(z == len(n)):
        print("matriks tersebut konsisten")
    else:
        print("matriks tersebut tidak konsisten")

cekConsisten(a)
cekConsisten(b)
cekConsisten(c)

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 merupakan
angka")
```

```python
                    break
                else:
                    x+=1
        if(x==y):
            print("semua isi matriks merupakan angka")


cekInt(a)
cekInt(b)
cekInt(c)

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


ordo(a)
ordo(b)
ordo(d)
ordo(e)

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
```

```python
    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")


jumlah(a,b)
jumlah(a,d)



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("dapat dikalikan")
        vwxy = [[0 for j in range(w)] for i in range(x)]
        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)
```

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

zz = [[1,2,3],[1,2,3]]
zx = [[1],[2],[3]]
kali(zz,zx)
kali(a,b)
kali(a,e)
kali(a,zx)

def DeterminanHitung(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 = DeterminanHitung(As)
                total += sign * A[0][fc] * sub_det
        else:
            return "tidak dapat dihitung determinan, bukan
matrix bujursangkar"
```
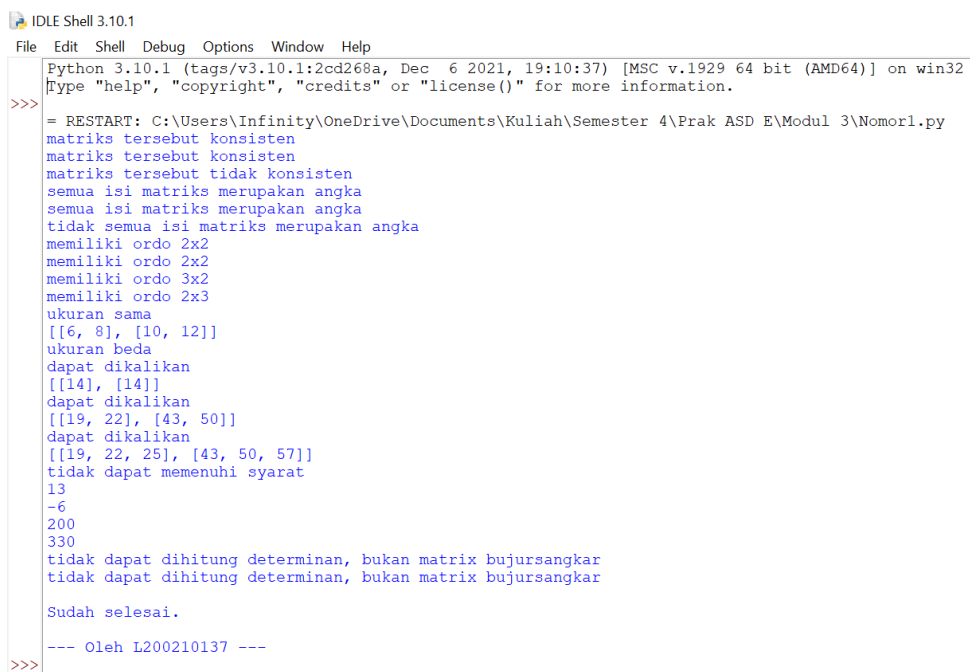
```python
    else:
        return "tidak dapat dihitung determinan, bukan matrix
bujursangkar"
    return total


z = [[3,1],[2,5]]
x = [[1,2,1],[3,3,1],[2,1,2]]
v = [[1,-2,0,0],[3,2,-3,1],[4,0,5,1],[2,3,-1,4]]
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]]
print(DeterminanHitung(z))
print(DeterminanHitung(x))
print(DeterminanHitung(v))
print(DeterminanHitung(r))
print(DeterminanHitung(d))
print(DeterminanHitung(e))
print('')
print('Sudah selesai.')


print('\n--- Oleh L200210137 ---')
```

IDLE Shell 3.10.1

File Edit Shell Debug Options Window Help

```
    Python 3.10.1 (tags/v3.10.1:2cd268a, Dec  6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    = RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul 3\Nomor1.py
    matriks tersebut konsisten
    matriks tersebut konsisten
    matriks tersebut tidak konsisten
    semua isi matriks merupakan angka
    semua isi matriks merupakan angka
    tidak semua isi matriks merupakan angka
    memiliki ordo 2x2
    memiliki ordo 2x2
    memiliki ordo 3x2
    memiliki ordo 2x3
    ukuran sama
    [[6, 8], [10, 12]]
    ukuran beda
    dapat dikalikan
    [[14], [14]]
    dapat dikalikan
    [[19, 22], [43, 50]]
    dapat dikalikan
    [[19, 22, 25], [43, 50, 57]]
    tidak dapat memenuhi syarat
    13
    -6
    200
    330
    tidak dapat dihitung determinan, bukan matrix bujursangkar
    tidak dapat dihitung determinan, bukan matrix bujursangkar

    Sudah selesai.

    --- Oleh L200210137 ---
>>>
```

Nomor 2 :

```python
#NO.2
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)])

buatNol(2,4)
buatNol(3)

def buatIdentitas(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)])

buatIdentitas(4)
buatIdentitas(2)
print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```

IDLE Shell 3.10.1

File   Edit   Shell   Debug   Options   Window   Help

```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec  6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul 3\Nomor2.py
membuat matriks 0 dengan ordo 2x4
[[0, 0, 0, 0], [0, 0, 0, 0]]
membuat matriks 0 dengan ordo 3x3
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
membuat matriks identitas dengan ordo4x4
[[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
membuat matriks identitas dengan ordo2x2
[[1, 0], [0, 1]]

Sudah selesai.

--- Oleh L200210137 ---
>>>
```

Nomor 3 :

```
#NO.3
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def pushAw(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node
    def pushAk(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 insert(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
            prev.next = node
            node.next = current
        return self.head
    def deleteNode(self, position):
        if self.head == None:
            return
        temp = self.head
```

```python
        if position == 0:
            self.head = temp.next
            temp = None
            return
        for i in range(position -1 ):
            temp = temp.next
            if temp is None:
                break
        if temp is None:
            return
        if temp.next is None:
            return
        next = temp.next.next
        temp.next = None
        temp.next = next
    def search(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.pushAw(21)
llist.pushAw(22)
llist.pushAw(12)
llist.pushAw(14)
llist.pushAw(2)
llist.pushAw(19)
llist.pushAk(9)
llist.deleteNode(0)
llist.insert(1,6)
print(llist.search(21))
print(llist.search(29))
llist.display()

print('')
print('\nSudah selesai.')
```

```
print('\n--- Oleh L200210137 ---')
```

File   Edit   Shell   Debug   Options   Window   Help

```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec  6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul 3\Nomor3.py
True
False
2 14 12 22 21 1 9

Sudah selesai.

--- Oleh L200210137 ---
>>>
```

Nomor 4 :

```
#NO.4
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)
print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```

IDLE Shell 3.10.1

File Edit Shell Debug Options Window Help

```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec  6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul 3\Nomor4.py
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

Sudah selesai.

--- Oleh L200210137 ---
>>>
```