Nama : Resky Budi Nugroho

NIM : L200180018

Kelas : A

Module 3 Tugas

1. Nomor 1

TUGAS1.py - C:/Users/ACER/Documents/GitHub/algostruk/MODUL_3/TUGAS1.py (3.8

File Edit Format Run Options Window Help

```
# 1a
def cetakMatriks(matriks):
    panjang = len(matriks)
    hasil = True
    for x in matriks:
        lebar = len(x)
        if lebar != panjang:
            hasil = False
            break
        for i in x:
            if type(i) != int:
                hasil = False
                break
    return hasil

m1 = [[1,2], [3,4]]
m2 = [[1,"s"],[3,4]]

print ("m1  =", cetakMatriks(m1))
print ("m2  =", cetakMatriks(m2))
```

Python 3.8.0 Shell

File Edit Shell Debug C

```
Python 3.8.0 (tags/v
tel)] on win32
Type "help", "copyri
>>>
====== RESTART: C:/U
m1  = True
m2  = False
>>>
```

```
a = [[1,2],[3,4]]
b = [[5,6],[7,8]]
c = [[12,3,"y"],[12,33,4]]
d = [[3,4],[2,4],[1,5]]

# 1a
def cetakMatriks(matriks):
    panjang = len(matriks)
    hasil = True
    for x in matriks:
        lebar = len(x)
        if lebar != panjang:
            hasil = False
            break
        for i in x:
            if type(i) != int:
                hasil = False
                break
    return hasil

#1B
def ukuran(n):
    x,y = 0,0
    for i in range(len(n)):
        x+=1
        y = len(n[i])
    print("mempunyai panjang "+str(x)+" dan lebar "+str(y))

ukuran(a)
ukuran(b)
ukuran(d)
```

Python 3.8.0 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23
tel)] on win32
Type "help", "copyright", "credits" or "license()" for m
>>>
====== RESTART: C:/Users/ACER/Documents/GitHub/algostruk
mempunyai panjang 2 dan lebar 2
mempunyai panjang 2 dan lebar 2
mempunyai panjang 3 dan lebar 2
>>>
```

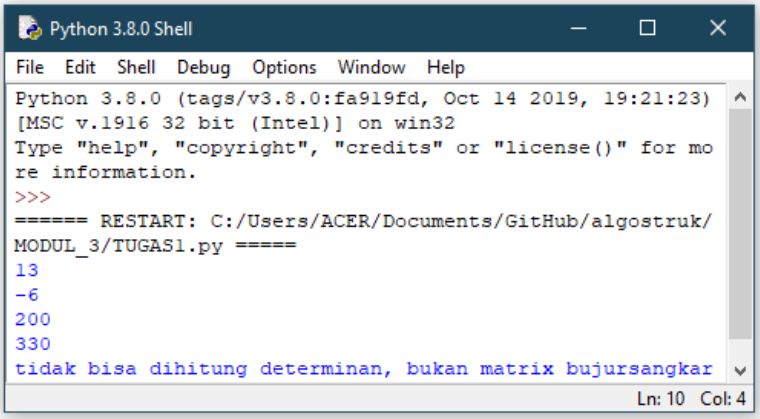
```python
#1C
def jumlah(n,m):
    x,y = 0,0
    for i in range(len(n)):
        x+=1
        y = len(n[i])
    xy = [[0 for j in range(x)] for i in range(y)]

    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("hasilnya adalah",xy)
    else:
        print("ukuran beda")

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

```
>>>
====== RESTART: C:/Users/ACER/Docu
mempunyai panjang 2 dan lebar 2
mempunyai panjang 2 dan lebar 2
mempunyai panjang 3 dan lebar 2
ukuran sama
hasilnya adalah [[6, 8], [10, 12]]
ukuran beda
>>>
```

```python
#1d
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)):
                    vwxy[i][j] += n[i][k] * m[k][j]
        print("hasilnya adalah =",vwxy)

    else:
        print("tidak memenuhi syarat")

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

```
tel)] on win32
Type "help", "copyright", "credits" or "l
>>>
====== RESTART: C:/Users/ACER/Documents/(
bisa dikalikan
[[0], [0]]
hasilnya adalah = [[14], [14]]
bisa dikalikan
[[0, 0], [0, 0]]
hasilnya adalah = [[19, 22], [43, 50]]
tidak memenuhi syarat
tidak memenuhi syarat
>>>
```

```
#le
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


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(determHitung(z))
print(determHitung(x))
print(determHitung(v))
print(determHitung(r))
print(determHitung(d))
```



2. Nomor 2

```
#2a
def buatNol(m,n= None):
    if n == None:
        n=m
    print([[0 for j in range (m)] for i in range (n)])
buatNol(2,3)
buatNol(3)

#2b
def buatIdentitas(n):
    print([[1 if j==i else 0 for j in range(n)] for i in range(n)])

buatIdentitas(4)
buatIdentitas(2)
```

```
Python 3.8.0 Shell
File  Edit  Shell  Debug  Options  Window  Help
>>>
====== RESTART: C:/Users/ACER/Documents/GitHub/algostruk/MODUL_3/tugas2.py ===
[[0, 0], [0, 0], [0, 0]]
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
[[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
[[1, 0], [0, 1]]
>>>
```

3. Nomor 3

tugas3.py - C:/Users/ACER/Documents/GitHub/algostruk/MODUL_3/tugas3.py (3.8.0)

File Edit Format Run Options Window Help

```
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node
#3c
    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

#3d
    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
#3e
    def hapus(self, position):
        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 tampil(self):
        current = self.head
        while current is not None:
            print(current.data, end = ' ')
            current = current.next
```

Python 3.8.0 Shell

File Edit Shell Debug Options Window Help

```
>>>
====== RESTART: C:/Users/ACER/Documents/GitHub/algostruk/MODUL_3
/tugas3.py =====
19 2 14 12 22 21 9 False
False
19 2 14 12 22 1 9
```

Ln: 17 Col: 4

4. Nomor 4

```
class Node:
    def __init__(self, data):
        self.data = data
        self.prev = None
class DNode:
    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
a = DNode()
a.awal(7)
a.awal(1)
a.akhir(6)
a.akhir(4)
a.printList(a.head)
```