

REYLIAN PREALDREAM ANAREKA
L200180087
KELAS D

```
Modul_3.py - D:\Kuliah\Praktikum Algoritma dan Struktur Data\Modul 3\Modul_3.py (37.3)
File Edit Format Run Options Window Help
print('NO 1')

print()
print('NO 1 A')
a = [[1,2],[3,4]]
b = [[5,6],[7,8]]
c = [[12,3,'a','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 cekKonsisten(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 konsisten")
    else:
        print("matrik tidak konsisten")

cekKonsisten(a)
cekKonsisten(b)
cekKonsisten(c)

print()
print('NO 1 B')
def cekKonsisten(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:
            z+=1
    if(z==y):
        print("semua isi matriks adalah angka")

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

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

cekOrdo(a)
cekOrdo(b)
cekOrdo(c)

Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (vs3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Kuliah\Praktikum Algoritma dan Struktur Data\Modul 3\Modul_3.py
NO 1
NO 1 A
matriks konsisten
matriks konsisten
matrik tidak konsisten

NO 1 B
semua isi matriks adalah angka
semua isi matriks adalah angka
tidak semua isi matriks adalah angka

NO 1 C
mempunyai ordo 2x2
mempunyai ordo 2x2
mempunyai ordo 3x2
mempunyai ordo 2x3
ukuran sama
[[6, 8], [10, 12]]
ukuran beda

NO 1 D
bisa dikalikan
[[14], [14]]
bisa dikalikan
[[19, 22], [43, 50]]
bisa dikalikan
[[19, 22, 25], [43, 50, 57]]
tidak memenuhi syarat

NO 1 E
13
-6
200
330
tidak bisa dihitung determinan, bukan matrik bujursangkar
tidak bisa dihitung determinan, bukan matrik bujursangkar

NO 2
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 ordo3x3
[[1, 0, 0], [0, 1, 0], [0, 0, 1]]
```

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:/Kuliah/Praktikum Algoritma dan Struktur Data/Modul 3/Modul_3.py
NO 1
NO 1 A
matriks konsisten
matriks konsisten
matrik tidak konsisten
NO 1 B
semua isi matriks adalah angka
semua isi matriks adalah angka
tidak semua isi matriks adalah angka
NO 1 C
mempunyai ordo 2x2
mempunyai ordo 2x2
mempunyai ordo 3x2
mempunyai ordo 2x3
ukuran sama
[[6, 8], [10, 12]]
ukuran beda
NO 1 D
bisa dikalikan
[[14], [14]]
bisa dikalikan
[[19, 22], [43, 50]]
bisa dikalikan
[[19, 22, 25], [43, 50, 57]]
tidak memenuhi syarat
NO 1 E
13
-6
200
330
tidak bisa dihitung determinan, bukan matriks bujursangkar
tidak bisa dihitung determinan, bukan matriks bujursangkar
NO 2
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 ordo 4x4
[[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
membuat matriks identitas dengan ordo 3x3
[[1, 0, 0], [0, 1, 0], [0, 0, 1]]
D:/Kuliah/Praktikum Algoritma dan Struktur Data/Modul 3/Modul_3.py (3,7,3)
File Edit Format Run Options Window Help
y = len(n[i])
print("mempunyai ordo "+str(x)+"*"+str(y))
print()
print("NO 1 C")
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
    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)
print()
print("NO 1 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("bisa dikalikan")
        aa = 0
        for i in range(len(n)):
            for j in range(len(n[i])):
                aa+=n[i][j]*m[j][0]
        print(aa)
    else:
        print("tidak bisa dikalikan")
kali(a,b)
kali(a,c)
kali(a,d)
kali(a,e)
kali(b,c)
kali(b,d)
kali(b,e)
kali(c,d)
kali(c,e)
kali(d,e)
kali(e,e)
kali(a,[1,2,3,4])
kali([1,2,3,4],a)
kali([1,2,3,4],[1,2,3,4])
kali([1,2,3,4],[1,2,3,4,5])
kali([1,2,3,4],[1,2,3,4,5,6])
kali([1,2,3,4],[1,2,3,4,5,6,7])
kali([1,2,3,4],[1,2,3,4,5,6,7,8])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43])
kali([1,2,3,4],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
```

```
Modul_3.py - D:/Kuliah/Praktikum Algoritma dan Struktur Data/Modul 3/Modul_3.py (3.7.3)
File Edit Format Run Options Window Help
vwx = [[0 for j in range(w)] for i in range(x)]
for i in range(len(m)):
    for j in range(len(m[0])):
        for k in range(len(m)):
            vwx[i][j] += m[i][k] * m[k][j]
    print(vwx)
else:
    print("tidak 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)

print()
print("NO 1 B")
def determinan(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)):
            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 = determinan(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]]
m = [[3,1],[2,5]]
lin 30 Col 9

Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
mempunyai ordo 2x3
ukuran sama
[[6, 8], [10, 12]]
ukuran beda

NO 1 D
bisa dikalikan
[[14], [14]]
bisa dikalikan
[[19, 22], [43, 50]]
bisa dikalikan
[[19, 22, 25], [43, 50, 57]]
tidak memenuhi syarat

NO 1 E
13
-6
200
330
tidak bisa dihitung determinan, bukan matrix bujursangkar
tidak bisa dihitung determinan, bukan matrix bujursangkar

NO 2
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 ordo 4x4
[[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
membuat matriks identitas dengan ordo 2x2
[[1, 0], [0, 1]]

NO 3
True
False
2 14 12 22 21 1 9

NO 4
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
>>>
```

Nomer 2

```
Modul_3.py - D:/Kuliah/Praktikum Algoritma dan Struktur Data/Modul 3/Modul_3.py (3.7.3)
File Edit Format Run Options Window Help
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(determinan(z))
print(determinan(x))
print(determinan(v))
print(determinan(r))
print(determinan(d))
print(determinan(e))

print("\nNO 2")
def buatMatriks(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))

buatMatriks(2,4)
buatMatriks(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("\nNO 3")
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def push(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
```


Nomer 4

```
Modul_3.py - D:/Kuliah/Praktikum Algoritma dan Struktur Data/Modul 3/Modul_3.py (3.7.3)
File Edit Format Run Options Window Help
#####
print(l1list.search(29))
l1list.display()

print('\n\nNO 4')
class Node:
    def __init__(self, data):
        self.data = data
        self.prev = None
        self.next = 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.prev = None
        if self.head is None:
            new_node.next = 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("% 3d" % (node.data))
            last = node
            node = node.next
        print("\nDari Belakang :")
        while(last is not None):
            print("% 3d" % (last.data))
            last = last.prev
l1list = DoublyLinkedList()
l1list.awal(7)
l1list.awal(1)
l1list.akhir(6)
l1list.akhir(4)
l1list.printlist(l1list.head)

Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
#####
bisa dikalikan
[[14], [14]]
bisa dikalikan
[[19, 22], [43, 50]]
bisa dikalikan
[[19, 22, 25], [43, 50, 57]]
tidak memenuhi syarat

NO 1 E
13
-6
200
330
tidak bisa dihitung determinan, bukan matrix bujursangkar
tidak bisa dihitung determinan, bukan matrix bujursangkar

NO 2
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 ordo 4x4
[[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
membuat matriks identitas dengan ordo 2x2
[[1, 0], [0, 1]]

NO 3
True
False
2 14 12 22 21 1 9

NO 4
menambah pada awal 7
menambah pada awal 1
menambah pada akhir 6
menambah pada akhir 4

Dari Depan :
 7
 1
 6
 4

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