

Nama : Bagus Zizou Satiaji

NIM : L200180212

Kelas : Praktikum Algoritma dan Struktur data H

## MODUL 3

### Collections, Arrays, and Linked Structures

#### Nomer 1

MODUL3.py - E:/KULIAH/semester4/PRAK\_ALGOSTRUK/MODUL 3/MODUL3.py (2.7.15)

File Edit Format Run Options Window Help

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

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])):
            c = type(n[i][j])
            if (c!=y):
                a = False
                break
        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")
```

```

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

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")

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(vwxy)

def hitungD(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

```


## Hasil

```
==== RESTART: E:/KULIAH/semester4/PRAK_ALGOSTRUK/MODUL 3/MODUL3.py =====
>>> cekKonsis(b)
matriks konsisten
>>> cekInt(d)
semua isi matriks adalah angka
>>> cekInt(c)
tidak semua isi matriks adalah angka
>>> ordo(a)
mempunyai ordo 2x2
>>> jumlah(d,e)
ukuran beda
>>> jumlah(a,b)
ukuran sama
[[20, 6], [15, 13]]
>>> jumlah(n,m)

>>> jumlah(c,e)
ukuran sama

--
>>> kali(d,e)
bisa dikalikan
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
[[19, 26, 33], [18, 24, 30], [21, 27, 33]]
>>> kali(a,d)
tidak memenuhi syarat
>>> hitungD(d)
'tidak bisa dihitung determinan, bukan matrix bujursangkar'
>>> hitungD(a)
14
>>> |
```

## Nomer 2

 MODUL3.py - E:/KULIAH/semester4/PRAK\_ALGOSTRUK/MODUL 3/MODUL3.py (2.7.15)

File Edit Format Run Options Window Help

```
#Nomer 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)])

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)])
```

## Hasil

```
==== RESTART: E:/KULIAH/semester4/PRAK_ALGOSTRUK/MODUL 3/MODUL3.py =====
>>> buatNol(3)
membuat matriks 0 dengan ordo 3x3
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
>>> buatNol(4)
membuat matriks 0 dengan ordo 4x4
[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
>>> buatIdentitas(4)
membuat matriks identitas dengan ordo4x4
[[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
>>> |
```

### Nomer 3

MODUL3.py - E:/KULIAH/semester4/PRAK\_ALGOSTRUK/MODUL 3/MODUL3.py (2.7.15)

File Edit Format Run Options Window Help

```
#Nomer 3
class Node(object):
    def __init__(self, data, next=None):
        self.data = data
        self.next = next

def MakeNode(list):
    a = Node(list[0])
    if len(list) > 1:
        b = a
        for i in range(1, len(list)):
            b.next = Node(list[i])
            b = b.next
    return a

def kunjungi(head):
    curNode = head
    while curNode != None:
        print(curNode.data)
        curNode = curNode.next

def cari(head, yang_dicari):
    temp = head
    while temp != None:
        if temp.data == yang_dicari:
            return temp
        temp = temp.next
    return Node(None)

def tambahDepan(head):
    temp = Node("tambah depan", head)
    return temp

def tambahAkhir(head):
    temp = head
    while temp.next != None:
        temp = temp.next
    temp.next = Node("tambah akhir")
    return head
```

```
def tambahAkhir(head):

def tambah(head, posisi):
    """ Menambahkan simpul sebelum posisi """
    temp = head
    while temp != None:
        if temp.next.data == posisi:
            temp_belakang = temp.next
            temp.next = Node("tambah tengah", temp_belakang)
            return head
        temp = temp.next
    return None

def hapus(head, posisi):
    temp = head
    while temp != None:
        if temp.next.data == posisi:
            temp_belakang = temp.next.next
            temp.next = temp_belakang
            return head
        temp = temp.next
    return None

a = MakeNode(["Bagus", "Zizou", "Satieji", "Zizo", "Zizu"])

print(a.data)
c = cari(a, "Bagus")
print(c.next.data)

print()
kunjungi(a)

print()
a = tambahDepan(a)
kunjungi(a)

print()
a = tambahAkhir(a)
kunjungi(a)

print()
a = tambah(a, "Bagus")
kunjungi(a)

print()
a = hapus(a, "Bagus")
kunjungi(a)
print("\n")
```

## Hasil

```
===== RESTART: E:/KULIAH/semester4/PRAK_ALGOSTRUK/MODUL 3/MODUL3.py =====
Bagus
Zizou
()
Bagus
Zizou
Satiaji
Zizo
Zizu
()
tambah depan
Bagus
Zizou
Satiaji
Zizo
Zizu
()
tambah depan
Bagus
Zizou
Satiaji
Zizo
Zizu
tambah akhir
()
tambah depan
tambah tengah
Bagus
Zizou
Satiaji
Zizo
Zizu
tambah akhir
()
tambah depan
tambah tengah
Zizou
Satiaji
Zizo
Zizu
tambah akhir
```

## Nomer 4

```
MODUL3.py - E:\KULIAH\semester4\PRAK_ALGOSTRUK\MODUL 3\MODUL3.py (3.7.0)
File Edit Format Run Options Window Help

#Nomer 4
class DNode(object):
    def __init__(self, data):
        self.data = data
        self.next = None
        self.prev = None

def massDNodeCreator(list):
    a = DNode(list[0])
    p = a
    for i in list[1:]:
        p.next = DNode(i)
        p.next.prev = p
        p = p.next
    return a

def tambahSimpulAwal(head, data):
    data = DNode(data)
    data.next = head
    data.next.prev = data
    return data

def tambahSimpulAkhir(head, data):
    data = DNode(data)
    temp = head
    while temp.next != None:
        temp = temp.next
    temp.next = data
    return head

list = ["e", "f", "g", "h"]
a = massDNodeCreator(list)
print(a.next.next.next.prev.prev.data)

a = tambahSimpulAwal(a, "awal")
print(a.next.prev.data)

a = tambahSimpulAkhir(a, "akhir")
print(a.next.next.next.next.next.data)

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\KULIAH\semester4\PRAK_ALGOSTRUK\MODUL 3\MODUL3.py =====
f
awal
akhir
>>> |
```