

Nama : Dwiky Nugraha

NIM : L200170154

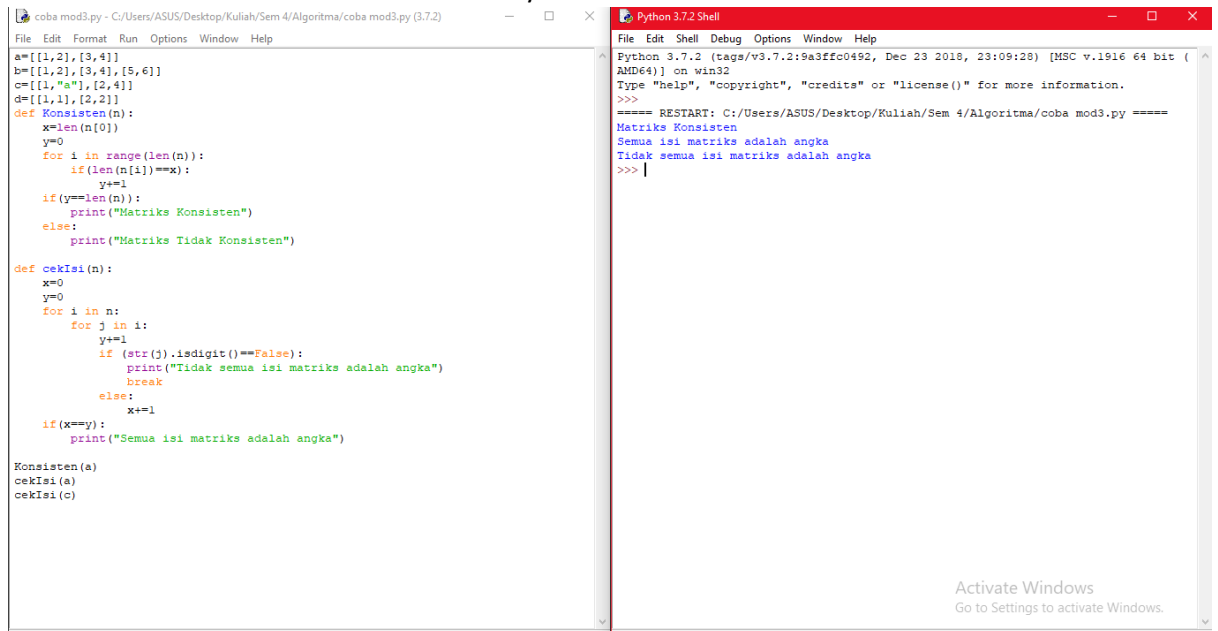
Kelas : D

Modul : 3

TUGAS

1. Array dua dimensi

- Memastikan bahwa isi dan ukuran matrix-nya konsisten



```
coba mod3.py - C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/coba mod3.py (3.7.2)
File Edit Format Run Options Window Help

a=[[1,2],[3,4]]
b=[[1,2],[3,4],[5,6]]
c=[[1,"a"],[2,4]]
d=[[1,1],[2,2]]
def Konsisten(n):
    x=len(n[0])
    y=0
    for i in range(len(n)):
        if(len(n[i])==x):
            y+=1
    if(y==len(n)):
        print("Matriks Konsisten")
    else:
        print("Matriks Tidak Konsisten")

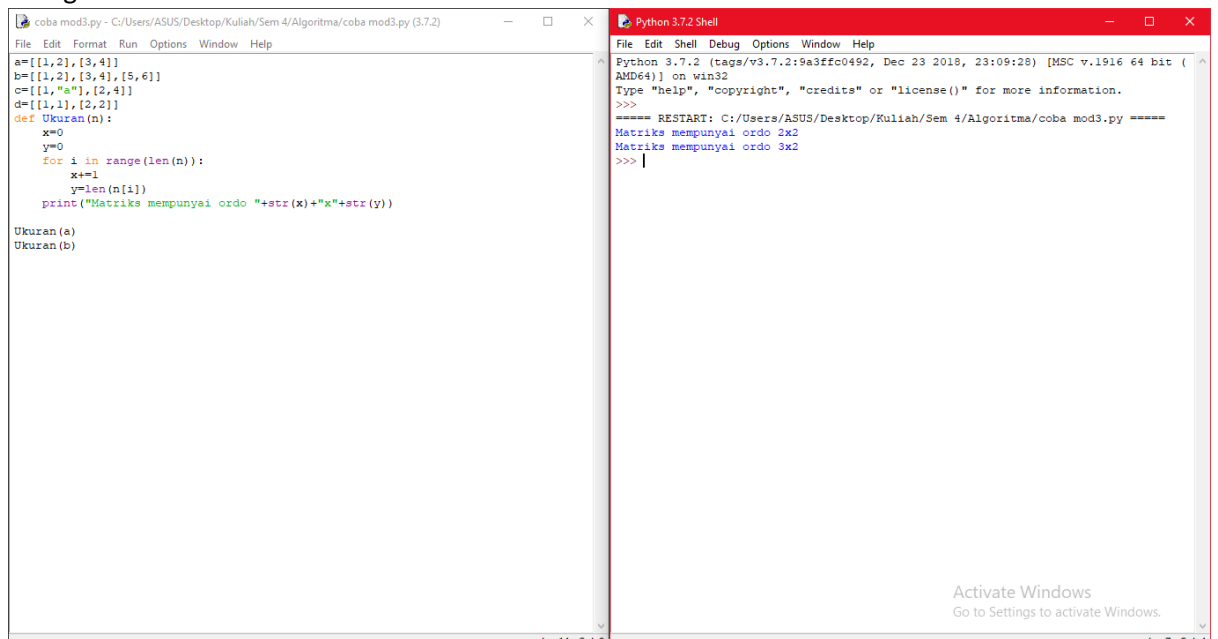
def cekIsi(n):
    x=0
    y=0
    for i in n:
        for j in i:
            y+=1
            if (str(j).isdigit()==False):
                print("Tidak semua isi matriks adalah angka")
                break
            else:
                x+=1
        if(x==y):
            print("Semua isi matriks adalah angka")

Konsisten(a)
cekIsi(a)
cekIsi(c)
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/coba mod3.py =====
Matriks Konsisten
Semua isi matriks adalah angka
Tidak semua isi matriks adalah angka
>>> |
```

- Mengambil ukuran matrix



```
coba mod3.py - C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/coba mod3.py (3.7.2)
File Edit Format Run Options Window Help

a=[[1,2],[3,4]]
b=[[1,2],[3,4],[5,6]]
c=[[1,"a"],[2,4]]
d=[[1,1],[2,2]]
def Ukuran(n):
    x=0
    y=0
    for i in range(len(n)):
        x+=1
        y=len(n[i])
        print("Matriks mempunyai ordo "+str(x)+"x"+str(y))

Ukuran(a)
Ukuran(b)
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/coba mod3.py =====
Matriks mempunyai ordo 2x2
Matriks mempunyai ordo 3x2
>>> |
```

- Menjumlahkan dua matrix

The screenshot shows a Python IDE with a file named `coba mod3.py`. The code defines a function `Jumlah(a, d)` that adds two matrices. It checks if the dimensions are the same and prints the result. The execution in the Python 3.7.2 Shell shows the output: `Ukuran matriks sama` and the resulting matrix `[[2, 3], [5, 6]]`.

```
File Edit Format Run Options Window Help
a=[[1,2],[3,4]]
b=[[1,2],[3,4],[5,6]]
c=[[1,"a"],[2,4]]
d=[[1,1],[2,2]]
def Jumlah(n,m):
    x=0
    y=0
    z=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)]
    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 matriks 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 matriks berbeda")

Jumlah(a,d)
```

```
Python 3.7.2 Shell
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/coba mod3.py =====
Ukuran matriks sama
[[2, 3], [5, 6]]
>>>
```

- Mengalikan dua matrix

The screenshot shows a Python IDE with a file named `coba mod3.py`. The code defines a function `Kali(n, m)` that multiplies two matrices. It checks if the dimensions are compatible and prints the result. The execution in the Python 3.7.2 Shell shows the output: `Matriks dapat dikalikan` and the resulting matrix `[[14], [20], [14]]`.

```
File Edit Format Run Options Window Help
a=[[1,2],[3,4]]
b=[[1,2],[3,4],[5,6]]
c=[[1,"a"],[2,4]]
d=[[1,1],[2,2]]
def Kali(n,m):
    a=0
    x=0
    y=0
    for i in range(len(n)):
        x+=1
        y=len(n[i])
    v=0
    w=0
    for i in range(len(m)):
        v+=1
        w=len(m[i])
    if (y==v):
        print("Matriks dapat dikalikan")
        hasil=[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)):
                    hasil[i][j]=n[i][k]*m[k][j]
        print(hasil)
    else:
        print("Matriks tidak memenuhi syarat")

xa=[[1,2,3],[2,3,4],[3,4,1]]
xb=[[1],[2],[3]]
Kali(xa,xb)
Kali(a,d)
```

```
Python 3.7.2 Shell
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/coba mod3.py =====
Matriks dapat dikalikan
[[14], [20], [14]]
Matriks dapat dikalikan
[[5, 5], [11, 11]]
>>>
```

- Menghitung determinan sebuah matrix bujursangkar

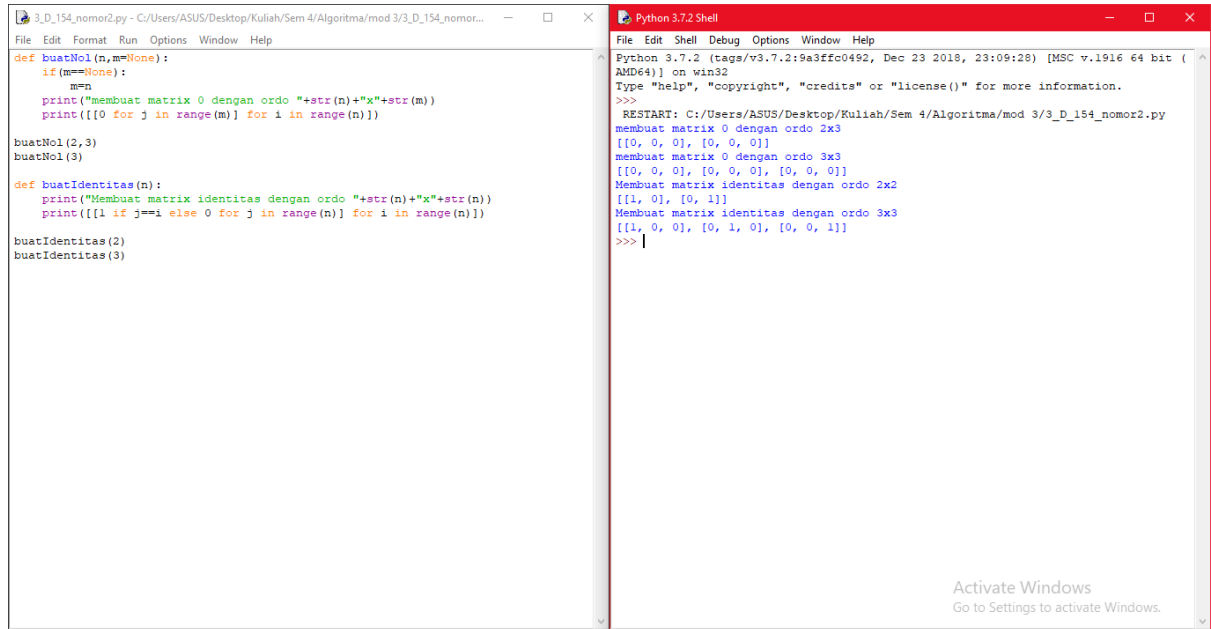
The screenshot shows a Python IDE with a file named `coba mod3.py`. The code defines a function `hitungDet(n, total=0)` that calculates the determinant of a square matrix using Laplace expansion. It checks if the matrix is square and prints the result. The execution in the Python 3.7.2 Shell shows the output: `Tidak dapat dihitung, bukan matrix bujursangkar`.

```
File Edit Format Run Options Window Help
a=[[1,2],[3,4]]
b=[[1,2],[3,4],[5,6]]
c=[[1,"a"],[2,4]]
d=[[1,1],[2,2]]
def hitungDet(n, total=0):
    x=len(n[0])
    y=0
    for i in range(len(n)):
        if (len(n[i])==x):
            y+=1
    if (y==len(n)):
        if (x==len(n)):
            indices=list(range(len(n)))
            if len(n)==2 and len(n[0])==2:
                val=n[0][0]*n[1][1]-n[1][0]*n[0][1]
                return val
            for i in indices:
                ns=n
                ns=ns[i:]
                height=len(ns)
                for i in range(height):
                    ns[i]=ns[i][0:i]+ns[i][i+1:]
                sign=(-1)**(i%2)
                sub_det=hitungDet(ns)
                total+=sign*n[0][i]*sub_det
            return "Tidak dapat dihitung, bukan matrix bujursangkar"
        else:
            return "Tidak dapat dihitung, bukan matrix bujursangkar"
    return total

print(hitungDet(a))
print(hitungDet(b))
```

```
Python 3.7.2 Shell
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/coba mod3.py =====
-2
Tidak dapat dihitung, bukan matrix bujursangkar
>>>
```

2. Matrix dan list comprehension. Buat matrix nol dan identitas



```
File Edit Format Run Options Window Help
def buatNol(n,m=None):
    if m==None:
        m=n
    print("membuat matrix 0 dengan ordo "+str(n)+"x"+str(m))
    print([[0 for j in range(m)] for i in range(n)])

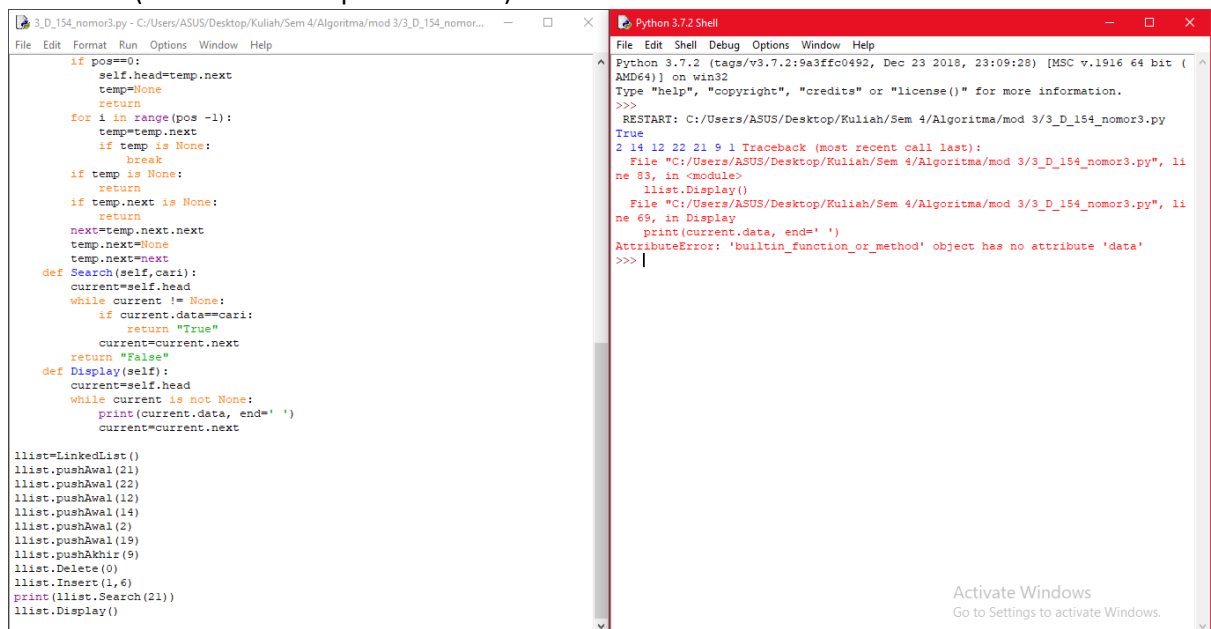
buatNol(2,3)
buatNol(3)

def buatIdentitas(n):
    print("Membuat matrix 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(2)
buatIdentitas(3)
```

```
Python 3.7.2 Shell
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/mod 3/3_D_154_nomor2.py
membuat matrix 0 dengan ordo 2x3
[[0, 0, 0], [0, 0, 0]]
membuat matrix 0 dengan ordo 3x3
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
Membuat matrix identitas dengan ordo 2x2
[[1, 0], [0, 1]]
Membuat matrix identitas dengan ordo 3x3
[[1, 0, 0], [0, 1, 0], [0, 0, 1]]
>>>
```

3. Linked list (belum bisa memperbaiki error)

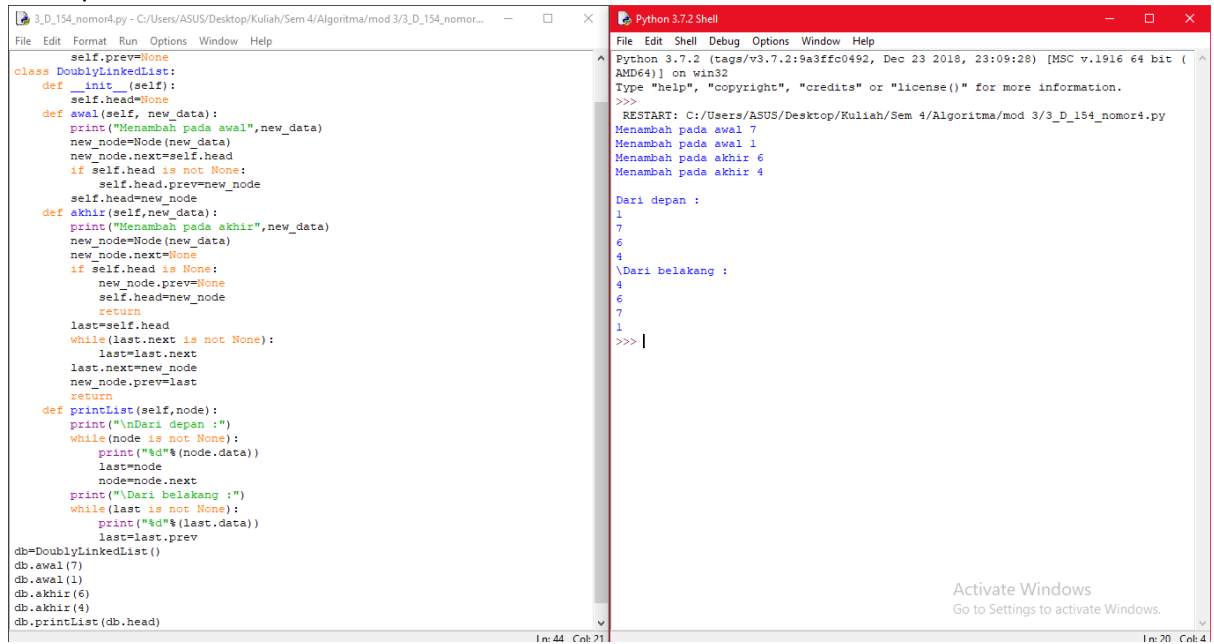


```
File Edit Format Run Options Window Help
if pos==0:
    self.head=temp.next
    temp=None
    return
for i in range(pos -1):
    temp=temp.next
    if temp is None:
        break
if temp is None:
    return
if temp.next is None:
    return
next=temp.next.next
temp.next=None
temp.next=temp
def Search(self,cari):
    current=self.head
    while current != None:
        if current.data==cari:
            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

l1list=LinkedList()
l1list.pushAwal(21)
l1list.pushAwal(22)
l1list.pushAwal(12)
l1list.pushAwal(14)
l1list.pushAwal(2)
l1list.pushAwal(19)
l1list.pushAkhir(9)
l1list.Delete(0)
l1list.Insert(1,6)
print(l1list.Search(21))
l1list.Display()
```

```
Python 3.7.2 Shell
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/mod 3/3_D_154_nomor3.py
2 14 12 22 21 9 1
Traceback (most recent call last):
  File "C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/mod 3/3_D_154_nomor3.py", line 83, in <module>
    l1list.Display()
  File "C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/mod 3/3_D_154_nomor3.py", line 69, in Display
    print(current.data, end=' ')
AttributeError: 'builtin_function_or_method' object has no attribute 'data'
>>>
```

4. Doubly linked list



The image shows a screenshot of a Python IDE with two windows. The left window is a code editor for a file named `3_D_154_nomor4.py`, and the right window is a Python 3.7.2 Shell showing the execution output.

Code Editor (Left Window):

```
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
db=DoublyLinkedList()
db.awal(7)
db.awal(1)
db.akhir(6)
db.akhir(4)
db.printList(db.head)
```

Python 3.7.2 Shell (Right Window):

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/ASUS/Desktop/Kuliah/Sem 4/Algoritma/mod 3/3_D_154_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
>>> |
```

At the bottom of the shell window, there is a watermark: "Activate Windows. Go to Settings to activate Windows."