

```
Python 3.7.7rc1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7rc1 (tags/v3.7.7rc1:93b7677f9c, Mar 4 2020, 12:05:18) [MSC v.1900 64 bi
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\Python37\algoritma
matriks konsisten
matrik tidak konsisten
matrik tidak konsisten
semua isi matriks adalah angka
semua isi matriks adalah angka
tidak semua isi matriks adalah angka
2
mempunyai ordo 2x2
2
mempunyai ordo 2x2
3
mempunyai ordo 3x3
ukuran sama
[[6, 8], [10, 12]]
ukuran beda
bisa dikalikan
[[0], [0]]
[[14], [14]]
bisa dikalikan
[[0, 0], [0, 0]]
[[19, 22], [43, 50]]
bisa dikalikan
[[0, 0, 0], [0, 0, 0]]
[[19, 22, 25], [43, 50, 57]]
tidak memenuhi syarat
13
-6
200
330
tidak bisa dihitung determinan, bukan matriks bujursangkar
tidak bisa dihitung determinan, bukan matriks bujursangkar
>>>
```

```
1.py - C:\Users\acer\AppData\Local\Programs\Python\Python37\Python37\algoritma_Mod...
File Edit Format Run Options Window Help

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)

def kali(n,m):
    as = 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")
        vvxy = [[0 for j in range(w)] for i in range(x)]

Ln: 15 Col: 0
Ln: 36 Col: 57
```

```
Python 3.7.7rc1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7rc1 (tags/v3.7.7rc1:93b7677f9c, Mar 4 2020, 12:05:18) [MSC v.1900 64 bi
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\Python37\algoritma
matriks konsisten
matrik tidak konsisten
matrik tidak konsisten
semua isi matriks adalah angka
semua isi matriks adalah angka
tidak semua isi matriks adalah angka
2
mempunyai ordo 2x2
2
mempunyai ordo 2x2
3
mempunyai ordo 3x3
ukuran sama
[[6, 8], [10, 12]]
ukuran beda
bisa dikalikan
[[0], [0]]
[[14], [14]]
bisa dikalikan
[[0, 0], [0, 0]]
[[19, 22], [43, 50]]
bisa dikalikan
[[0, 0, 0], [0, 0, 0]]
[[19, 22, 25], [43, 50, 57]]
tidak memenuhi syarat
13
-6
200
330
tidak bisa dihitung determinan, bukan matriks bujursangkar
tidak bisa dihitung determinan, bukan matriks bujursangkar
>>>
```

```
1.py - C:\Users\acer\AppData\Local\Programs\Python\Python37\Python37\algoritma_Mod...
File Edit Format Run Options Window Help

def kali(n,m):
    as = 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")
        vvxy = [[0 for j in range(w)] for i in range(x)]
        print(vvxy)
        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])
                    vvxy[i][j] += n[i][k] * m[k][j]
        print(vvxy)
    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)

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)):

Ln: 15 Col: 0
Ln: 36 Col: 57
```

```
Python 3.7.7rc1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7rc1 (tags/v3.7.7rc1:93b7677f9c, Mar 4 2020, 12:05:18) [MSC v.1900 64 bit
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\Python37\200180135_algostruk
matriks konsisten
matrik tidak konsisten
matrik tidak konsisten
semua isi matriks adalah angka
semua isi matriks adalah angka
tidak semua isi matriks adalah angka
2
mempunyai ordo 2x2
2
mempunyai ordo 2x2
3
mempunyai ordo 3x3
ukuran sama
[[6, 8], [10, 12]]
ukuran beda
bisa dikalikan
[[0], [0]]
bisa dikalikan
[[14], [14]]
bisa dikalikan
[[0, 0], [0, 0]]
[[19, 22], [43, 50]]
bisa dikalikan
[[0, 0, 0], [0, 0, 0]]
[[19, 22, 25], [43, 50, 57]]
tidak memenuhi syarat
13
-6
200
330
tidak bisa dihitung determinan, bukan matriks bujursangkar
tidak bisa dihitung determinan, bukan matriks bujursangkar
>>>
```

```
1.py - C:\Users\acer\AppData\Local\Programs\Python\Python37\200180135_algostruk_Mod...
File Edit Format Run Options Window Help

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 matriks bujursangkar"
        else:
            return "tidak bisa dihitung determinan, bukan matriks 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]]

Ln: 15 Col: 0
Ln: 36 Col: 57
19:39
18/03/2020
```

```
Python 3.7.7rc1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7rc1 (tags/v3.7.7rc1:93b7677f9c, Mar 4 2020, 12:05:18) [MSC v.1900 64 bit
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\Python37\200180135_algostruk
matriks konsisten
matrik tidak konsisten
matrik tidak konsisten
semua isi matriks adalah angka
semua isi matriks adalah angka
tidak semua isi matriks adalah angka
2
mempunyai ordo 2x2
2
mempunyai ordo 2x2
3
mempunyai ordo 3x3
ukuran sama
[[6, 8], [10, 12]]
ukuran beda
bisa dikalikan
[[0], [0]]
bisa dikalikan
[[14], [14]]
bisa dikalikan
[[0, 0], [0, 0]]
[[19, 22], [43, 50]]
bisa dikalikan
[[0, 0, 0], [0, 0, 0]]
[[19, 22, 25], [43, 50, 57]]
tidak memenuhi syarat
13
-6
200
330
tidak bisa dihitung determinan, bukan matriks bujursangkar
tidak bisa dihitung determinan, bukan matriks bujursangkar
>>>
```

```
1.py - C:\Users\acer\AppData\Local\Programs\Python\Python37\200180135_algostruk_Mod...
File Edit Format Run Options Window Help

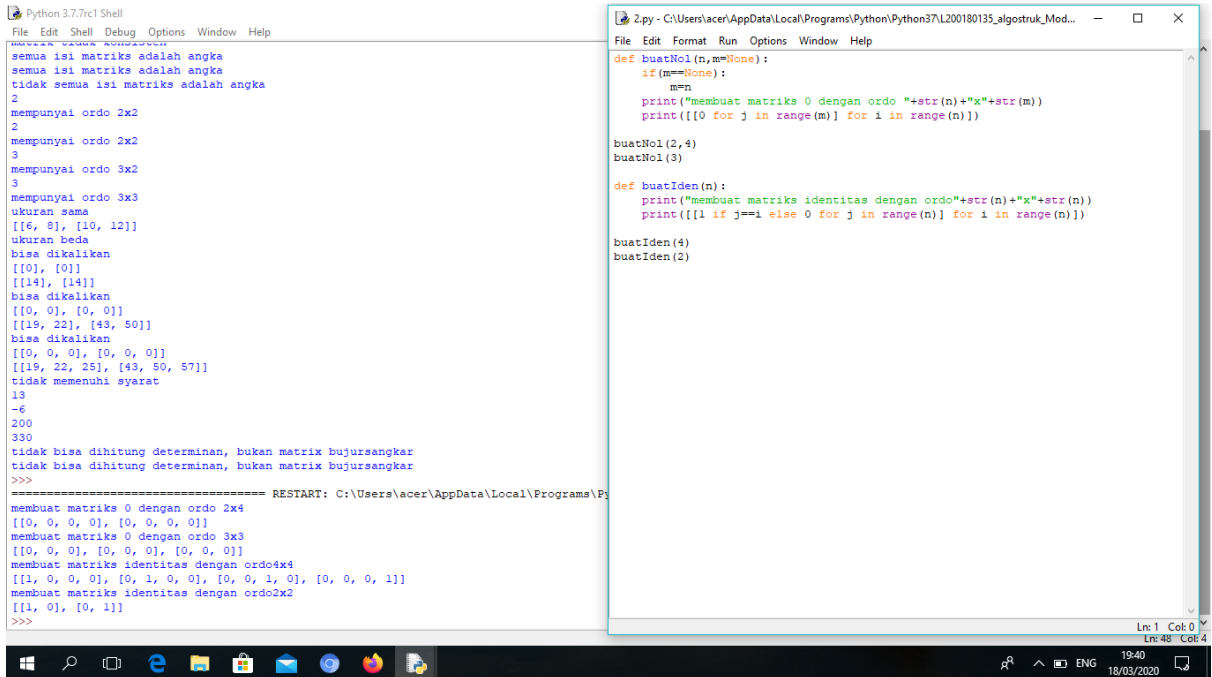
if(z == len(A)):
    if(x==len(A)):
        indices = list(range(len(A)))
        if len(A) == 2 and len(A[0]) == 2:
            val = A[0][0] * A[1][1] - A[1][0] * A[0][1]
            return val
        for fc in indices:
            As = A
            As = As[1:]
            height = len(As)
            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 matriks bujursangkar"
    else:
        return "tidak bisa dihitung determinan, bukan matriks 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))
print(determHitung(e))

Ln: 15 Col: 0
Ln: 36 Col: 57
19:39
18/03/2020
```

## 2. Nomor 2



The image shows two side-by-side Python IDE windows. The left window, titled 'Python 3.7.7rc1 Shell', contains a script that checks if a matrix is square, calculates its determinant, and creates identity matrices. The right window, titled '2.py - C:\Users\acer\AppData\Local\Programs\Python\Python37\...', contains functions to create zero and identity matrices.

```
Python 3.7.7rc1 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\...
semua isi matriks adalah angka
semua isi matriks adalah angka
tidak semua isi matriks adalah angka
2
mempunyai ordo 2x2
2
mempunyai ordo 2x2
3
mempunyai ordo 3x2
3
mempunyai ordo 3x3
ukuran sama
[[6, 8], [10, 12]]
ukuran beda
bisa dikalikan
[[0], [0]]
[[14], [14]]
bisa dikalikan
[[0, 0], [0, 0]]
[[19, 22], [43, 50]]
bisa dikalikan
[[0, 0, 0], [0, 0, 0]]
[[19, 22, 25], [43, 50, 57]]
tidak memenuhi syarat
13
-6
200
330
tidak bisa dihitung determinan, bukan matrik bujursangkar
tidak bisa dihitung determinan, bukan matrik bujursangkar
>>>
===== RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\...
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]]
>>>

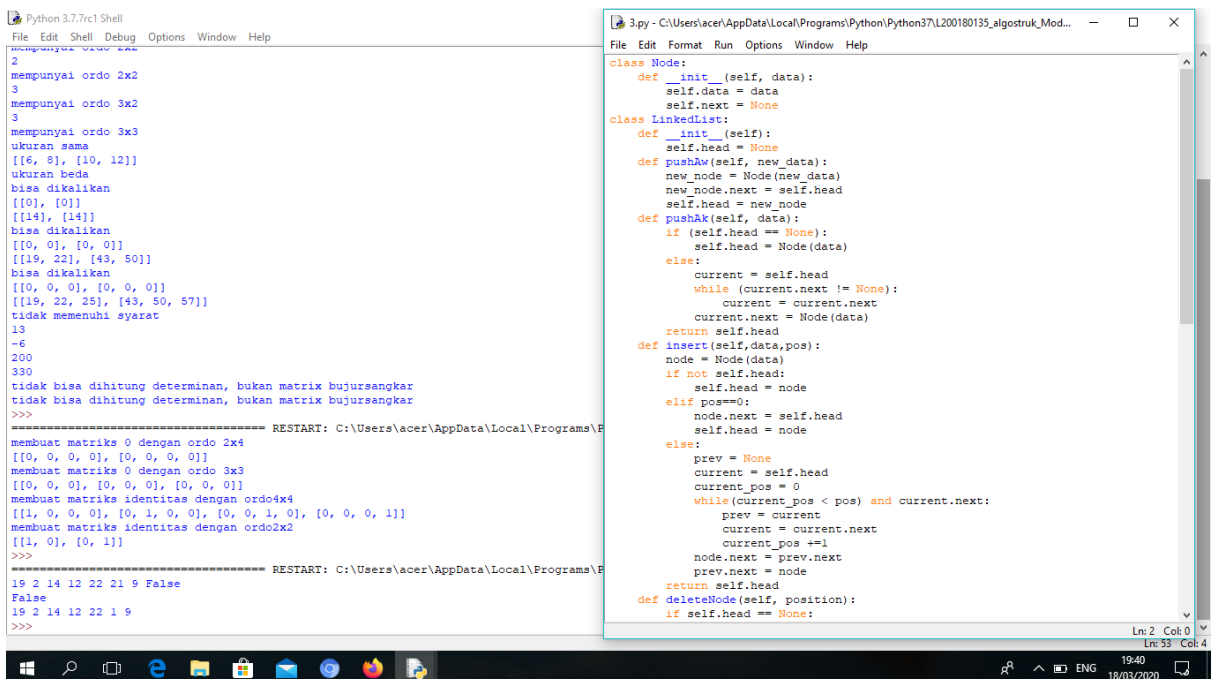
2.py - C:\Users\acer\AppData\Local\Programs\Python\Python37\...
File Edit Format Run Options Window Help
def buatNol(n,m=None):
    if m==None:
        m=n
    print("membuat matriks 0 dengan ordo "+str(n)+"x"+str(m))
    print([[0 for j in range(m)] for i in range(n)])

buatNol(2,4)
buatNol(3)

def buatIden(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)])

buatIden(4)
buatIden(2)
```

## 3. Nomor 3



The image shows two side-by-side Python IDE windows. The left window, titled 'Python 3.7.7rc1 Shell', contains a script that checks if a matrix is square, calculates its determinant, and creates identity matrices. The right window, titled '3.py - C:\Users\acer\AppData\Local\Programs\Python\Python37\...', contains a linked list implementation with Node and LinkedList classes.

```
Python 3.7.7rc1 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\...
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]]
>>>
===== RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\...
19 2 14 12 22 21 9 False
False
19 2 14 12 22 1 9
>>>

3.py - C:\Users\acer\AppData\Local\Programs\Python\Python37\...
File Edit Format Run Options Window Help
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
            node.next = prev.next
            prev.next = node
        return self.head
    def deleteNode(self, position):
        if self.head == None:
```

```
Python 3.7.7rc1 Shell
File Edit Shell Debug Options Window Help
mempunyai ordo 2x2
2
mempunyai ordo 3x2
3
mempunyai ordo 3x3
ukuran sama
[[6, 8], [10, 12]]
ukuran beda
bisa dikalikan
[[0], [0]]
[[14], [14]]
bisa dikalikan
[[0, 0], [0, 0]]
[[19, 22], [43, 50]]
bisa dikalikan
[[0, 0, 0], [0, 0, 0]]
[[19, 22, 25], [43, 50, 57]]
tidak memenuhi syarat
13
-6
200
330
tidak bisa dihitung determinan, bukan matriks bujursangkar
tidak bisa dihitung determinan, bukan matriks bujursangkar
>>>
===== RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\Python.exe
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]]
>>>
===== RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\Python.exe
19 2 14 12 22 21 9 False
False
19 2 14 12 22 1 9
>>>

3.py - C:\Users\acer\AppData\Local\Programs\Python\Python37\Python.exe
File Edit Format Run Options Window Help
def deleteNode(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 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

l1list = LinkedList()
l1list.pushAw(21)
l1list.pushAw(22)
l1list.pushAw(12)
l1list.pushAw(14)
l1list.pushAw(2)
l1list.pushAw(19)
Ln: 2 Col: 0
Ln: 53 Col: 4
19:40
18/03/2020
```

#### 4. Nomor 4

```
Python 3.7.7rc1 Shell
File Edit Shell Debug Options Window Help
tidak memenuhi syarat
13
-6
200
330
tidak bisa dihitung determinan, bukan matriks bujursangkar
tidak bisa dihitung determinan, bukan matriks bujursangkar
>>>
===== RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\Python.exe
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]]
>>>
===== RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\Python.exe
19 2 14 12 22 21 9 False
False
19 2 14 12 22 1 9
>>>
===== RESTART: C:\Users\acer\AppData\Local\Programs\Python\Python37\Python.exe
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
>>>

4.py - C:\Users\acer\AppData\Local\Programs\Python\Python37\Python.exe
File Edit Format Run Options Window Help
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.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
l1list = DoublyLinkedList()
l1list.awal(7)
Ln: 1 Col: 0
Ln: 71 Col: 4
19:41
18/03/2020
```

