Nama : Sang Aji Indutoro
NIM : L200180003

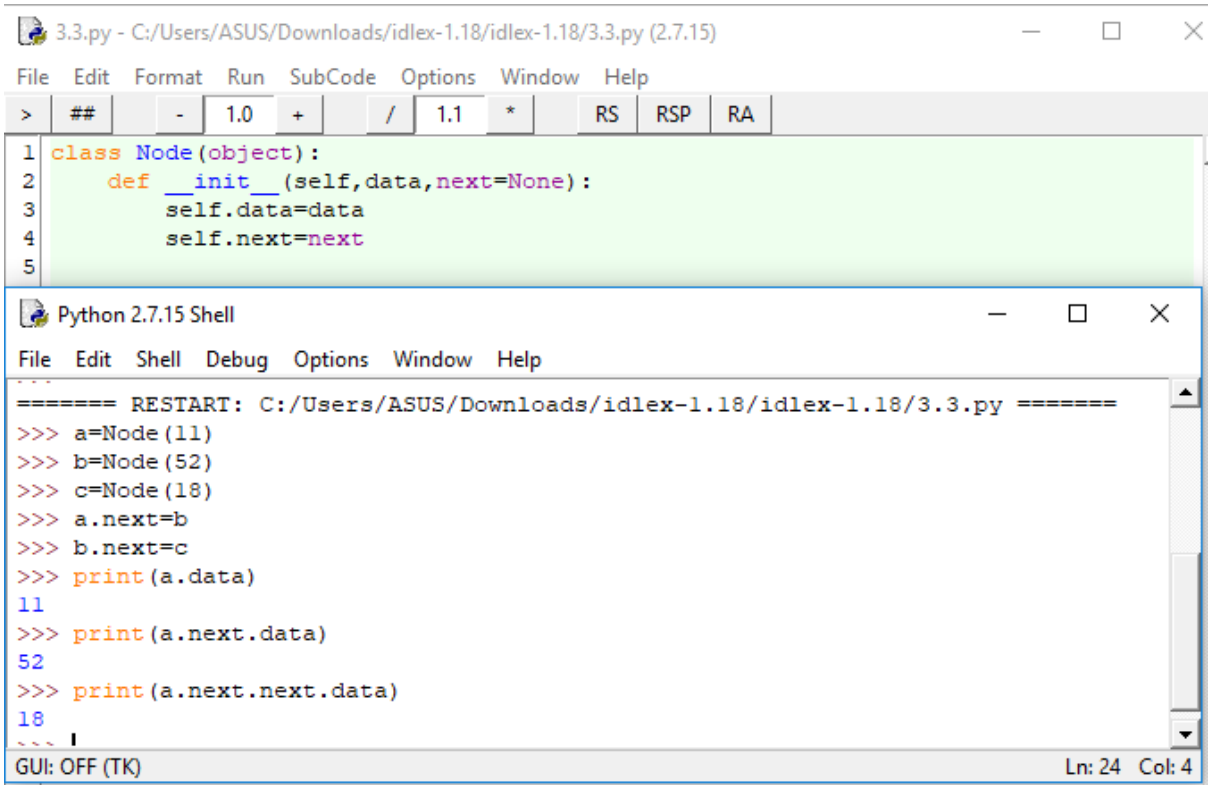
Latihan 3.1

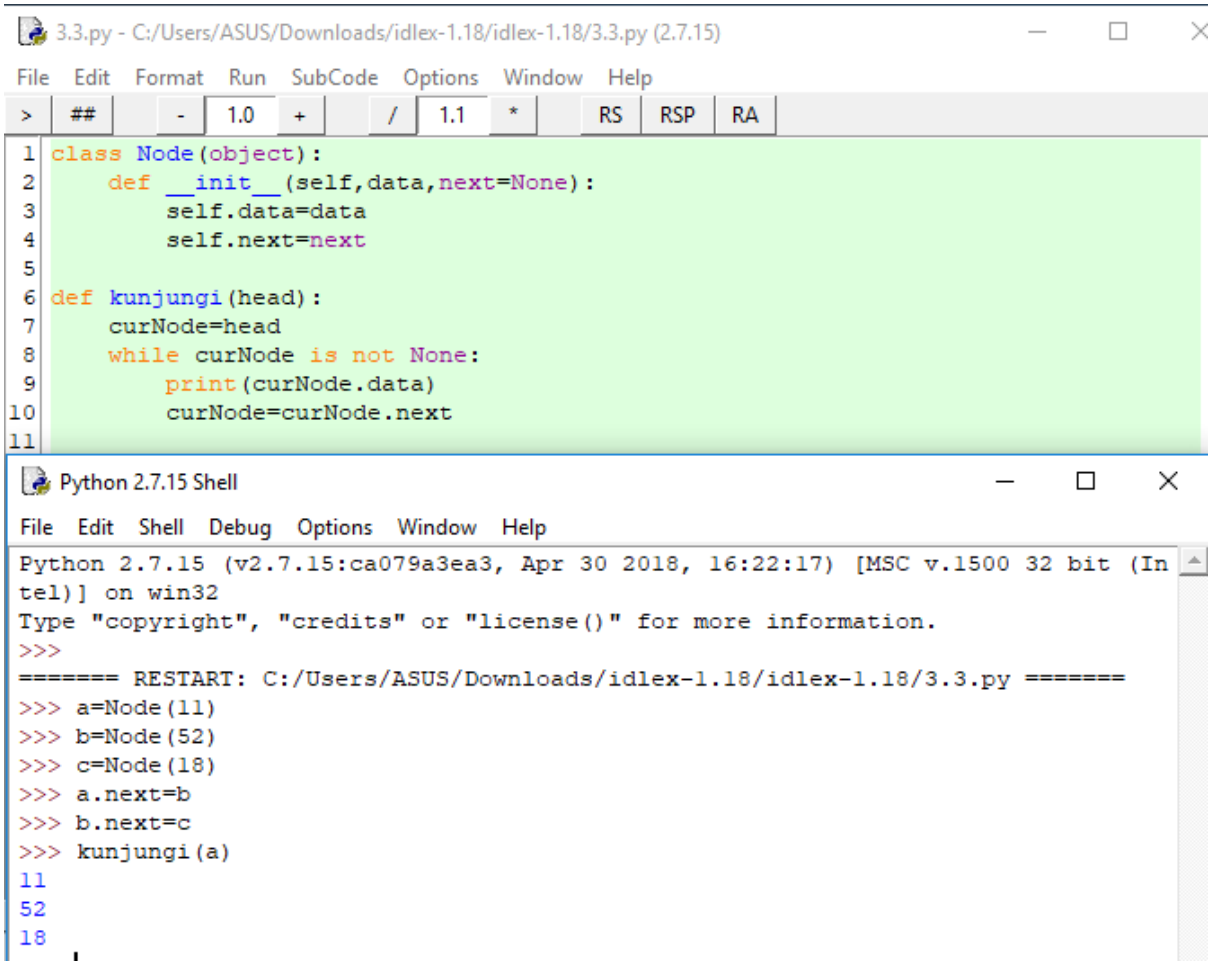
```
>>> A=[ [2,3],[5,7] ]
>>> A[0][1]
3
>>> A[1][1]
7
```

Latihan 3.2

```
>>> B=[ [0 for j in range(3)] for i in range(3)]
>>> B
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
>>>
```

Latihan 3.3




3.3.py - C:/Users/ASUS/Downloads/idlex-1.18/idlex-1.18/3.3.py (2.7.15)
File Edit Format Run SubCode Options Window Help
> ## - 1.0 + / 1.1 * RS RSP RA
1 class Node(object):
2 def __init__(self,data,next=None):
3 self.data=data
4 self.next=next
5
6 def kunjungi(head):
7 curNode=head
8 while curNode is not None:
9 print(curNode.data)
10 curNode=curNode.next
11
Python 2.7.15 Shell
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:22:17) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:/Users/ASUS/Downloads/idlex-1.18/idlex-1.18/3.3.py =======
>>> a=Node(11)
>>> b=Node(52)
>>> c=Node(18)
>>> a.next=b
>>> b.next=c
>>> kunjungi(a)
11
52
18

3.3.py - C:/Users/ASUS/Downloads/idlex-1.18/idlex-1.18/3.3.py (2.7.15)

File Edit Format Run SubCode Options Window Help

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

Python 2.7.15 Shell

File Edit Shell Debug Options Window Help

```
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:22:17
tel)] on win32
Type "copyright", "credits" or "license()" for more info
>>>
======= RESTART: C:/Users/ASUS/Downloads/idlex-1.18/idle
>>> a=DNode(11)
>>> b=DNode(52)
>>> c=DNode(18)
>>> a.next=b
>>> c.prev=b
>>> print(a.data)
11
>>> print(a.next.data)
52
>>> print(c.prev.data)
52
>>>
```

Soal-soal untuk Mahasiswa

1. Array dua dimensi, matrix yang berisi angka-angka.
   a. Fungsi cek isi dan ukuran matrix

```
def matrik(n):
    panjang=len(n)
    hasil=True
    for x in n:
        lebar=len(x)
        if lebar != panjang:
            hasil= False
            break
    for i in x:
        if type(i) != int:
            hasil = False
            break
    return hasil

def cek(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")
```

File Edit Shell Debug Options Window Help

```
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr
tel)] on win32
Type "copyright", "credits" or "licens
>>>
======= RESTART: C:/Users/ASUS/Downloa
>>> m1=[[2,3],[2,3]]
>>> m2=[[1,2,'yes'],[1,2.3]]
>>> matrik(m1)
True
>>> matrik(m2)
False
>>> cek(m1)
semua isi matriks adalah angka
>>> cek(m2)
Tidak semua isi matriks adalah angka
Tidak semua isi matriks adalah angka
>>>
```

b. Fungsi mengambil ukuran matrix

```
30 def ordo(n):
31     x,y = 0,0
32     for i in range(len(n)):
33         x+=1
34         y = len(n[i])
35     print("mempunyai ordo "+str(x)+"x"+str(y))
36
```

Code Browser Ln: 33 Col

```
>>> ordo(m1)
mempunyai ordo 2x2
```

c. Fungsi menjumlahkan matrix

```
37 def jumlah(n,m):
38     x,y = 0,0
39     for i in range(len(n)):
40         x+=1
41         y = len(n[i])
42     xy = [[0 for j in range(x)] for i in range(y)]
43
44     z = 0
45     if(len(n)==len(m)):
46         for i in range(len(n)):
47             if(len(n[i]) == len(m[i])):
48                 z+=1
49     if(z==len(n) and z==len(m)):
50         print("ukuran sama")
51         for i in range(len(n)):
52             for j in range(len(n[i])):
53                 xy[i][j] = n[i][j] + m[i][j]
54         print(xy)
55     else:
56         print("ukuran beda")
57
58 def kali(n,m):
```

Code Browser Ln: 38

```
>>> m3=[[1,2],[3,4]]
>>> m4=[[5,6],[7,8]]
>>> jumlah(m3,m4)
ukuran sama
[[6, 8], [10, 12]]
```

d. Fungsi menhitung determinan matrix

```
81  def det(A, total=0):
82      x = len(A[0])
83      z = 0
84      for i in range(len(A)):
85          if (len(A[i]) == x):
86              z+=1
87      if(z == len(A)):
88          if(x==len(A)):
89              indices = list(range(len(A)))
90              if len(A) == 2 and len(A[0]) == 2:
91                  val = A[0][0] * A[1][1] - A[1][0] * A[0][1]
92                  return val
93              for fc in indices:
94                  As = A
95                  As = As[1:]
96                  height = len(As)
97                  for i in range(height):
98                      As[i] = As[i][0:fc] + As[i][fc+1:]
99                  sign = (-1) ** (fc % 2)
100                 sub_det = determHitung(As)
101                 total += sign * A[0][fc] * sub_det
102         else:
103             return "tidak bisa dihitung determinan, bukan matrix bujursangkar"
104     else:
105         return "tidak bisa dihitung determinan, bukan matrix bujursangkar"
106     return total
107
```

Code Browser Ln: 89

```
>>> det(m3)
-2
>>> det(m4)
-2
>>>
```

2. Terkait matrix dan list comprehension

```
110 def buatNol(n,m=None):
111     if(m==None):
112         m=n
113     print("membuat matriks 0 dengan ordo "+str(n)+"x"+str(m))
114     print([[0 for j in range(m)] for i in range(n)])
115
116 def buatIden(n):
117     print("membuat matriks identitas dengan ordo"+str(n)+"x"+str(n))
118     print([[1 if j==i else 0 for j in range(n)] for i in range(n)])
```

Code Browser Ln: 116

```
>>> buatNol(2,3)
membuat matriks 0 dengan ordo 2x3
[[0, 0, 0], [0, 0, 0]]
>>> buatIden(3)
membuat matriks identitas dengan ordo3x3
[[1, 0, 0], [0, 1, 0], [0, 0, 1]]
>>>
```

3. Terkait linked list

```
125 class LinkedList:
126     def __init__(self):
127         self.head = None
128     def tambahDepan(self, new_data):
129         new_node = Node(new_data)
130         new_node.next = self.head
131         self.head = new_node
132     def tambahAkhir(self, data):
133         if (self.head == None):
134             self.head = Node(data)
135         else:
136             current = self.head
137             while (current.next != None):
138                 current = current.next
139             current.next = Node(data)
140         return self.head
141     def tambah(self,data,pos):
142         node = Node(data)
143         if not self.head:
144             self.head = node
145         elif pos==0:
146             node.next = self.head
147             self.head = node
148         else:
149             prev = None
150             current = self.head
151             current_pos = 0
152             while(current_pos < pos) and current.next:
153                 prev = current
154                 current = current.next
155                 current_pos +=1
156             prev.next = node
157             node.next = current
158         return self.head
159     def hapus(self, position):
160         if self.head == None:
161             return
162         temp = self.head
```

```
File  Edit  Shell  Debug  Options  Window  Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:22:17) [MSC v.1500 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:/Users/ASUS/Downloads/idlex-1.18/idlex-1.18/tgs3.py =======
>>> list=LinkedList()
>>> list.tambahDepan(15)
>>> list.tambahDepan(14)
>>> list.tambahDepan(13)
>>> list.tambahDepan(12)
>>> list.tambahAkhir(69)
<__main__.Node instance at 0x031A65D0>
>>> list.hapus(0)
>>> list.tambah(1,2)
<__main__.Node instance at 0x028A2B70>
>>> list.cari(14)
'True'
>>>
```

```
163         if position == 0:
164             self.head = temp.next
165             temp = None
166             return
167         for i in range(position -1 ):
168             temp = temp.next
169             if temp is None:
170                 break
171         if temp is None:
172             return
173         if temp.next is None:
174             return
175         next = temp.next.next
176         temp.next = None
177         temp.next = next
178     def cari(self, x):
179         current = self.head
180         while current != None:
181             if current.data == x:
182                 return "True"
183             current = current.next
184         return "False"
185     def display(self):
186         current = self.head
187         while current is not None:
188             current = current.next
189
190
```

Code Browser Ln: 137 Col: 31

4. Terkait doubly linked list

```
class Node:
    def __init__(self, data):
        self.data = data
        self.prev = 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
```

```
'True'
>>>
======= RESTART: C:/Users/ASUS/Do
>>> list=DoublyLinkedList()
>>> list.awal(1)
('menambah pada awal', 1)
>>> list.awal(2)
('menambah pada awal', 2)
>>> list.awal(3)
('menambah pada awal', 3)
>>> list.akhir(9)
('menambah pada akhir', 9)
>>> list.akhir(8)
('menambah pada akhir', 8)
>>> list.printList(list.head)

Dari Depan :
  3
  2
  1
  9
  8

Dari Belakang :
  8
  9
  1
  2
  3
>>>
GUI: OFF (TK)
```