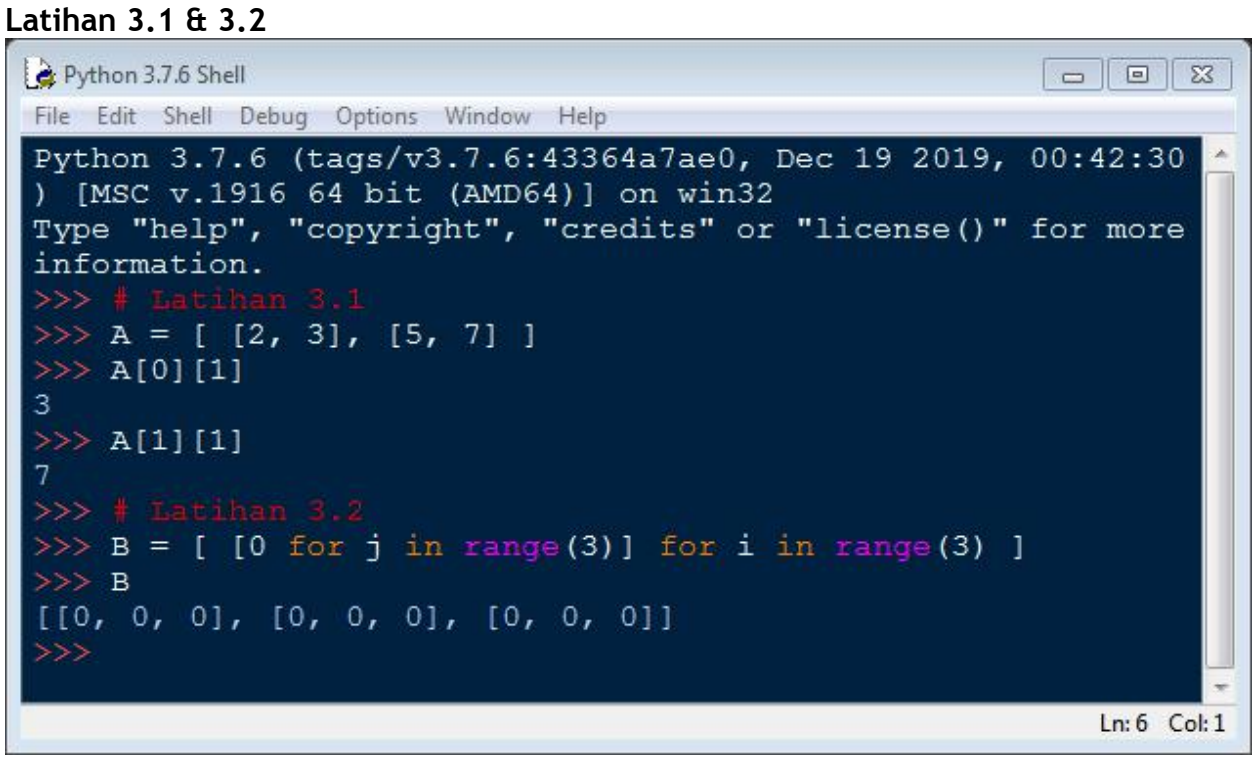
Laporan Praktikum Algoritma dan Struktur Data

NIM : L200180014

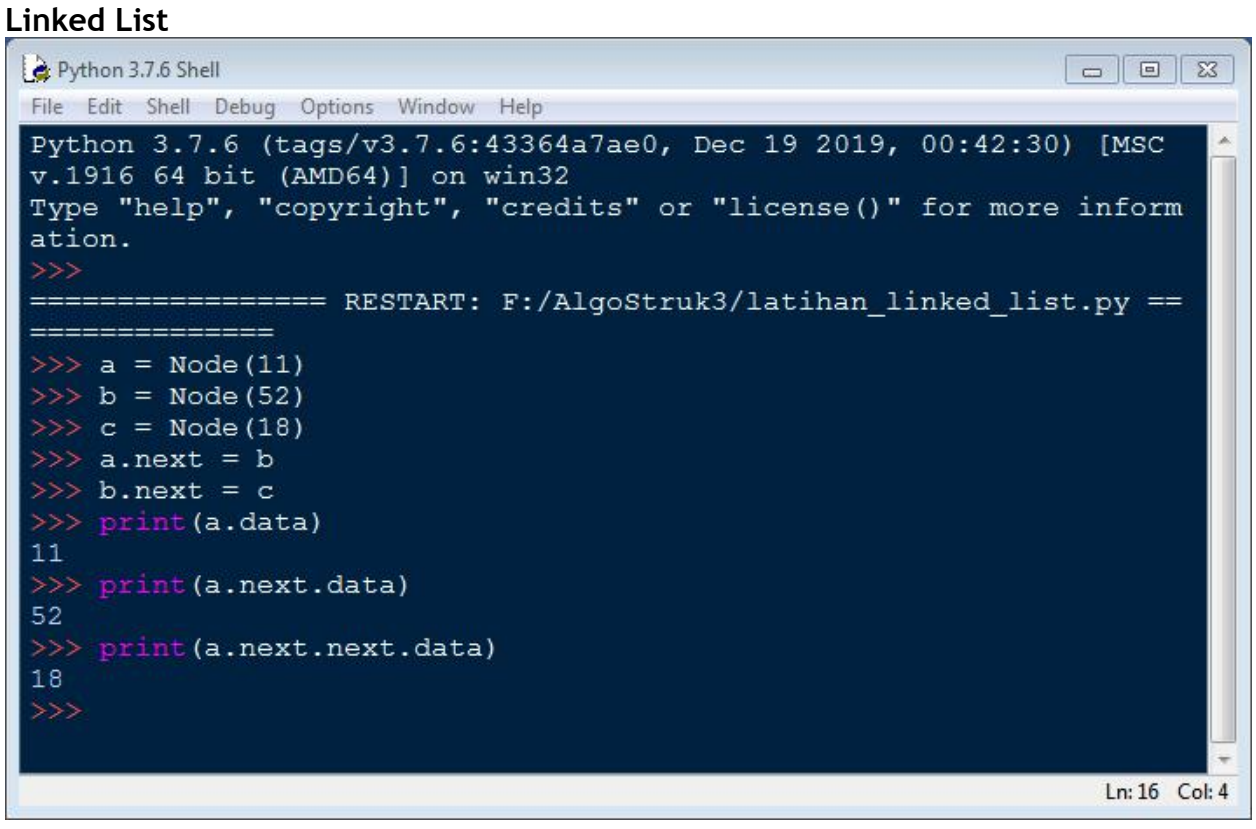
Nama : Andika Wirapala F. A.

Modul : 3

Latihan 3.1 & 3.2



Linked List



Mengunjungi Setiap Simpul dari Depan

latihan_linked_list.py - F:/AlgoStruk3/latihan_linked_list.py (3.7.6)

File Edit Format Run Options Window Help

```
class Node(object):
    """ Sebuah simpul di linked list """
    def __init__(self, data, next=None):
        self.data = data
        self.next = next

# Mengunjungi Setiap Simpul dari Depan
def kunjungi(head):
    curNode = head
    while curNode is not None:
        print(curNode.data)
        curNode = curNode.next
```

Python 3.7.6 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: F:/AlgoStruk3/latihan_linked_list.py ================
>>> a = Node(11)
>>> b = Node(52)
>>> c = Node(18)
>>> a.next = b
>>> b.next = c
>>> kunjungi(a)
11
52
18
>>>
```

Ln: 14 Col: 4

Soal-soal

1a) Cek Matriks

kode [~/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/3/kode] - .../1a.py - PyCharm

VCS Window Help

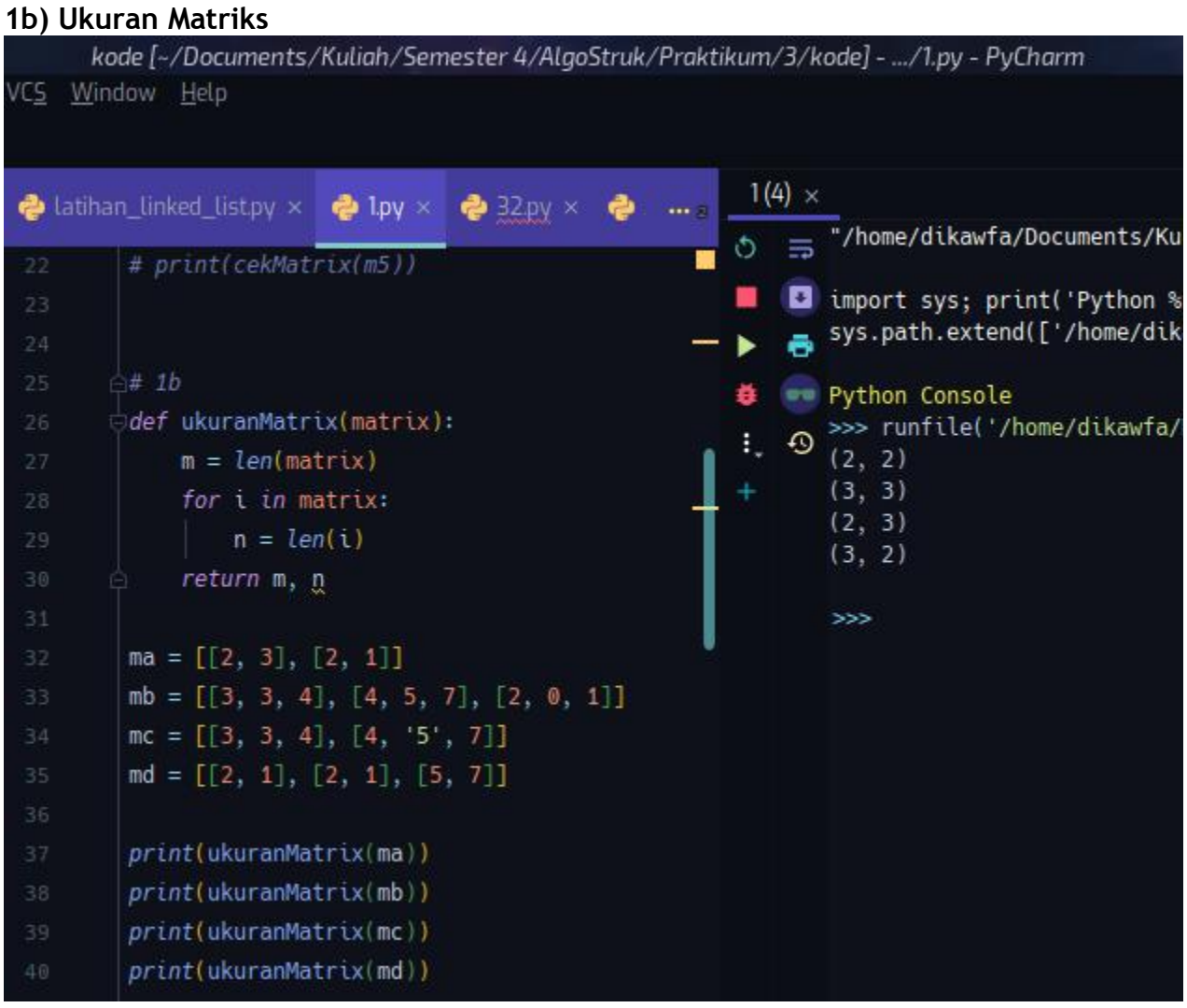
1a.py

```
# 1
def cekMatrix(matrix):
    panjang = len(matrix)
    for x in matrix:
        lebar = len(x)
        for i in x:
            if type(i) != int:
                break
    return panjang == lebar and type(i) == int


m1 = [[2, 3], [2, 1]]
m2 = [[3, 3, 4], [4, 5, 7], [2, 0, 1]]
m3 = [[3, 3, 4], [4, '5', 7], [2, 0, '1']]  # ada string
m4 = [[2, 1], [2, 3, 1]]  # beda ukuran
m5 = [['5', 3, 5], [5, 6, 5]]  # ada string

print(cekMatrix(m1))
print(cekMatrix(m2))
print(cekMatrix(m3))
print(cekMatrix(m4))
print(cekMatrix(m5))
```

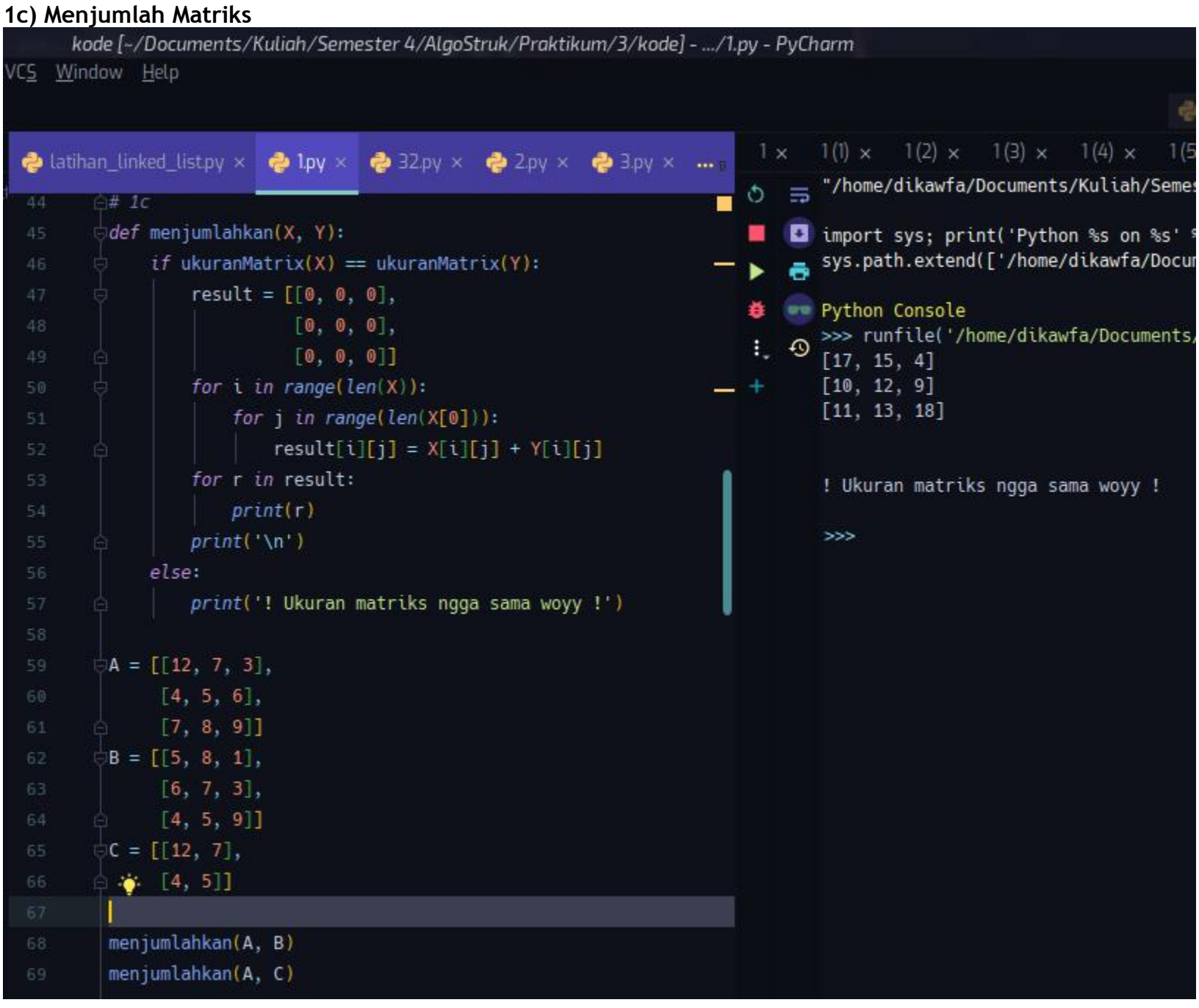
Run: 1a

```
"/home/dikawfa/Documents/Kuliah/Semester 4/
True
True
False
False
False

Process finished with exit code 0
```

1b) Ukuran Matriks



1c) Menjumlah Matriks



1d) Mengalikan Matriks

```python
# 1d
def mengalikan(X, Y):
    if barisY == kolomX:
        result = [[sum(a * b for a, b in zip(X_row, Y_col))
                   for Y_col in zip(*Y)] for X_row in X]

        for r in result:
            print(r)
    else:
        print('! Ukuran matrix tidak sesuai (baris Y != kolom X) !')
    print('\n')

X = [[1, 7, 3],
     [4, 5, 6],
     [7, 8, 9]]
Y = [[5, 0, 1, 2],
     [6, 7, 3, 0],
     [4, 5, 9, 1]]
Z = [[1, 2],
     [2, 5]]
mengalikan(X, Y)
mengalikan(X, Z)
```

```
"/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStruk/Pra
import sys; print('Python %s on %s' % (sys.version, sys.
sys.path.extend(['/home/dikawfa/Documents/Kuliah/Semeste
Python Console
>>> runfile('/home/dikawfa/Documents/Kuliah/Semester 4/A
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]

! Ukuran matrix tidak sesuai (baris Y != kolom X) !

>>>
```

1e) Determinan Matriks

```python
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):
        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 = 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
```

```
"/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/3/kod
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStru
Python Console
>>> runfile('/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStruk/Pra
>>> J = [[3, 1], [2, 5]]
... K = [[1, 2, 1], [3, 3, 1], [2, 1, 2]]
... L = [[1, -2, 0, 0], [3, 2, -3, 1], [4, 0, 5, 1], [2, 3, -1, 4]]
... M = [[3, 4], [2, 4], [1, 5]]
... print(determinan(J))
... print(determinan(K))
... print(determinan(L))
... print(determinan(M))
...
13
-6
200
tidak bisa dihitung determinan, bukan matrix bujursangkar

>>>
```

2)

latihan_linked_list.py × 1.py × 2.py × 3.py × 4.py ×

```python
# 2a
def buatNol(m, n=0):
    if n != 0:
        matriks = [[0 for i in range(m)] for j in range(n)]
        for item in matriks:
            print(item)
    else:
        matriks = [[0 for i in range(m)] for j in range(m)]
        for item in matriks:
            print(item)
    print('\n')

buatNol(3)
buatNol(2, 4)


# 2b
def buatIdentitas(m):
    matriks = [[1 if j == i else 0 for j in range(m)] for i in range(m)]
    for baris in matriks:
        print(baris)
    print('\n')

buatIdentitas(4)
buatIdentitas(3)
```

2 (1) ×

```
"/home/dikawfa/Docu
import sys; print('
sys.path.extend(['/
Python Console
>>> runfile('/home/
[0, 0, 0]
[0, 0, 0]
[0, 0, 0]


[0, 0, 0]
[0, 0, 0]
[0, 0, 0]
[0, 0, 0]


[1, 0, 0, 0]
[0, 1, 0, 0]
[0, 0, 1, 0]
[0, 0, 0, 1]



[1, 0, 0]
[0, 1, 0]
[0, 0, 1]


>>>
```

3)

latihan_linked_list.py | 1.py | 2.py | 3.py | 4.py

```python
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 is not None:
        print(curNode.data)
        curNode = curNode.next

def cari(head, yang_dicari):
    temp = head
    while temp is not None:
        if temp.data == yang_dicari:
            return temp
        temp = temp.next
    return Node(None)
```

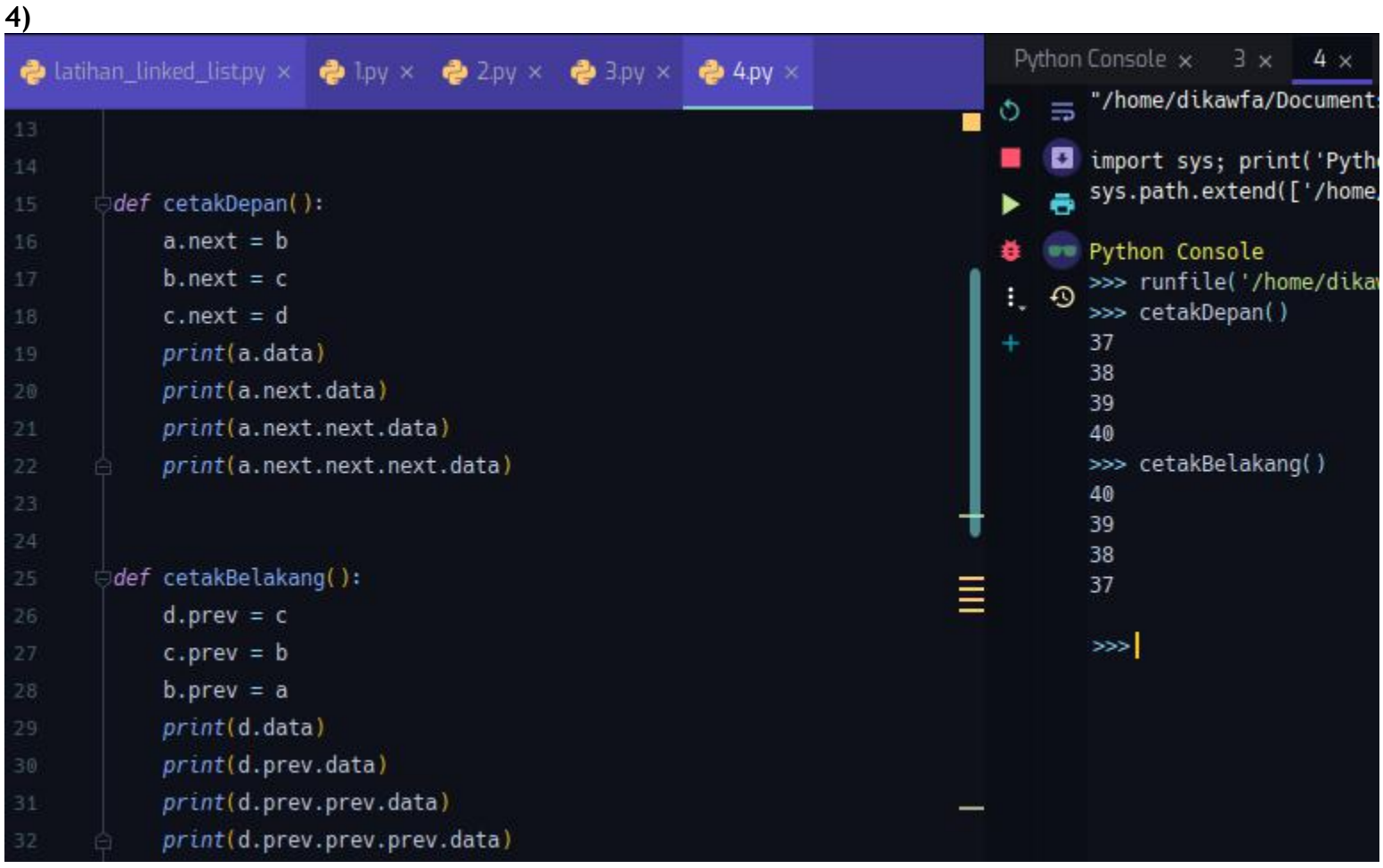
latihan_linked_list.py | 1.py | 2.py | 3.py | 4.py

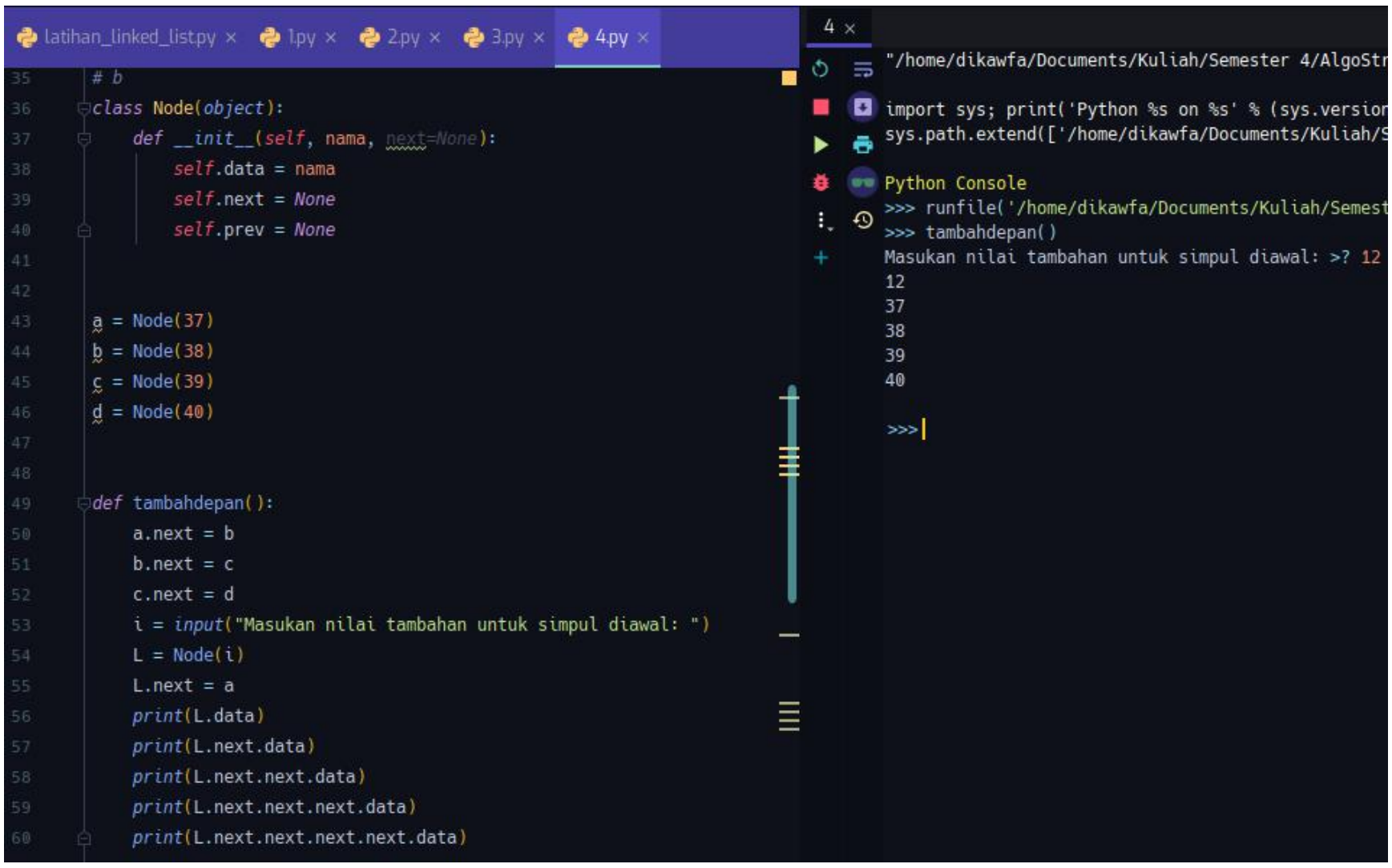
```python

def tambahdepan(head):
    temp = Node('tambah depan', head)
    return temp


def tambahAkhir(head):
    temp = head
    while temp.next is not None:
        temp = temp.next
    temp.next = Node('tambah akhir')
    return head


def tambah(head, posisi):
    """Menambahkan simpul sebelum posisi"""
    temp = head
    while temp is not None:
        if temp.next.data == posisi:
            temp_belakang = temp.next
            temp.next = Node('tambah tengah', temp_belakang)
            return head
        temp = temp.next
    return None


```

latihan_linked_list.py × 1.py × 2.py × 3.py × 4.py ×

```
        temp = temp.next
    return None


def hapus(head, posisi):
    temp = head
    while temp is not None:
        if temp.next.data == posisi:
            temp_belakang = temp.next.next
            temp.next = temp_belakang
            return head
        temp = temp.next
    return head

a = Node(1)
b = Node(2)
c = Node(3)
d = Node(4)
a.next = b
b.next = c
c.next = d

```

4)


latihan_linked_list.py
1.py
2.py
3.py
4.py
Python Console
3
4
"/home/dikawfa/Document
import sys; print('Pyth
sys.path.extend(['/home,
Python Console
>>> runfile('/home/dika
>>> cetakDepan()
37
38
39
40
>>> cetakBelakang()
40
39
38
37
>>>
def cetakDepan():
a.next = b
b.next = c
c.next = d
print(a.data)
print(a.next.data)
print(a.next.next.data)
print(a.next.next.next.data)
def cetakBelakang():
d.prev = c
c.prev = b
b.prev = a
print(d.data)
print(d.prev.data)
print(d.prev.prev.data)
print(d.prev.prev.prev.data)


latihan_linked_list.py
1.py
2.py
3.py
4.py
4
"/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStr
import sys; print('Python %s on %s' % (sys.version
sys.path.extend(['/home/dikawfa/Documents/Kuliah/S
Python Console
>>> runfile('/home/dikawfa/Documents/Kuliah/Semest
>>> tambahdepan()
Masukan nilai tambahan untuk simpul diawal: >? 12
12
37
38
39
40
>>>
# b
class Node(object):
def __init__(self, nama, next=None):
self.data = nama
self.next = None
self.prev = None
a = Node(37)
b = Node(38)
c = Node(39)
d = Node(40)
def tambahdepan():
a.next = b
b.next = c
c.next = d
i = input("Masukan nilai tambahan untuk simpul diawal: ")
L = Node(i)
L.next = a
print(L.data)
print(L.next.data)
print(L.next.next.data)
print(L.next.next.next.data)
print(L.next.next.next.next.data)

latihan_linked_list.py × 1.py × 2.py × 3.py × 4.py ×

```python

# c
class Node(object):
    def __init__(self, nama, next=None):
        self.data = nama
        self.next = None
        self.prev = None


def tambahAkhir():
    i = input("Masukan nilai tambahan untuk simpul diakhir: ")
    a = Node(37)
    b = Node(38)
    c = Node(39)
    d = Node(40)
    L = Node(i)
    d.prev = c
    c.prev = b
    b.prev = a
    L.prev = d
    print(L.prev.prev.prev.prev.data)
    print(L.prev.prev.prev.data)
    print(L.prev.prev.data)
    print(L.prev.data)
    print(L.data)

```

4 ×

```
"/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStru

import sys; print('Python %s on %s' % (sys.version,
sys.path.extend(['/home/dikawfa/Documents/Kuliah/Se

Python Console
>>> runfile('/home/dikawfa/Documents/Kuliah/Semeste
>>> tambahdepan()
Masukan nilai tambahan untuk simpul diawal: >? 12
12
37
38
39
40
>>> tambahAkhir()
Masukan nilai tambahan untuk simpul diakhir: >? 10
37
38
39
40
10

>>>
```