

Nama : Rifqi Alwan

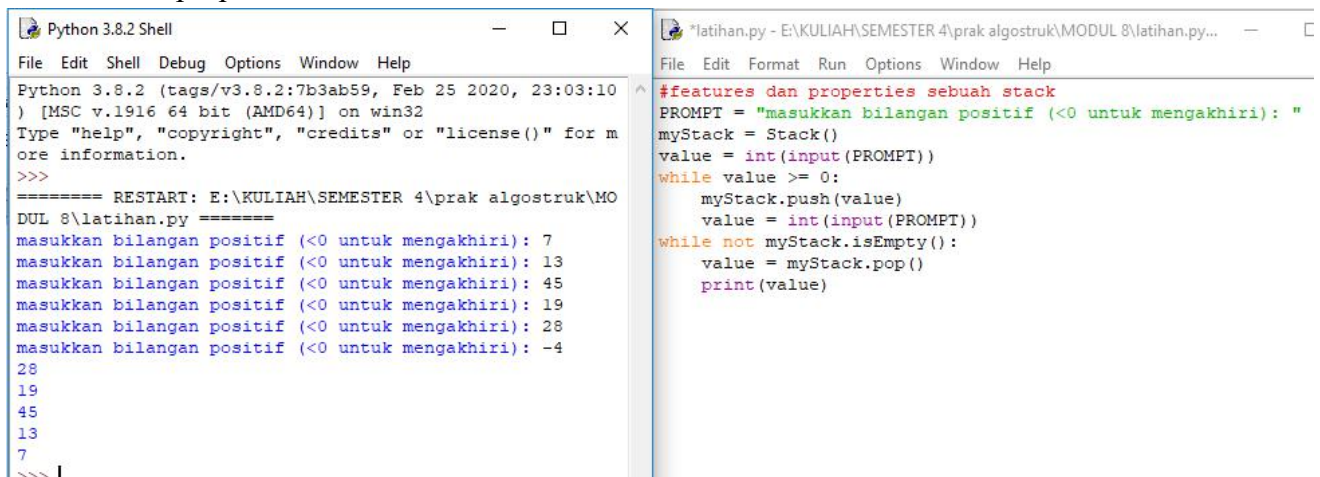
NIM : L200180008

Kelas A

## LATIHAN MODUL 8

### STACKS

#### 1. Features dan properties sebuah stack



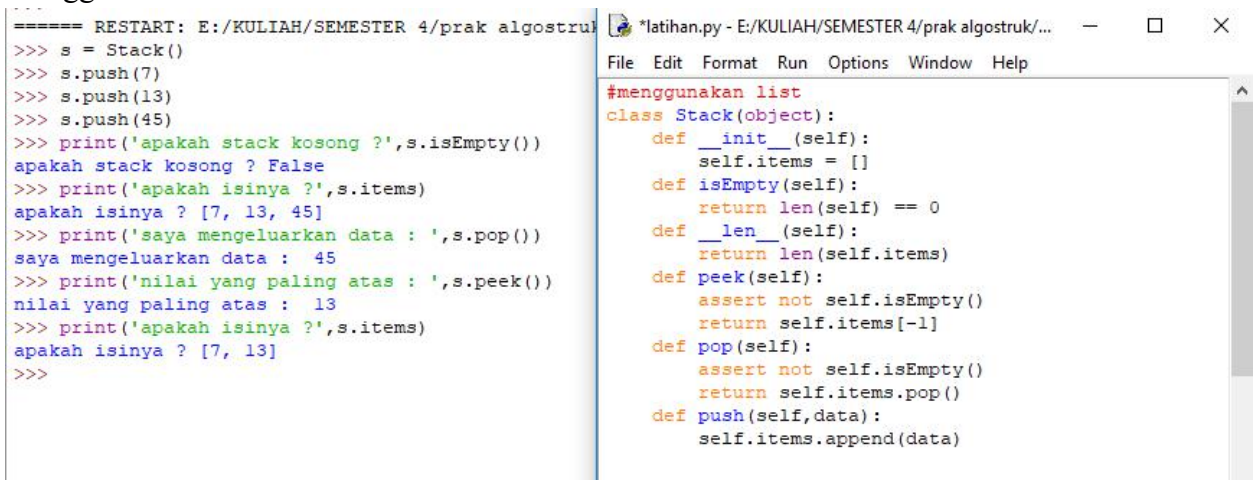
The image shows two side-by-side windows. The left window is a Python 3.8.2 Shell with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). It displays the output of a script where a stack is used to store positive integers. The user enters values 7, 13, 45, 19, 28, and -4. The stack stores 7, 13, 45, 19, and 28. The right window shows the source code of the script, titled '\*latihan.py - E:\KULIAH\SEMESTER 4\prak algostruk\MODUL 8\latihan.py...'. The code defines a Stack class and uses it to push and pop values.

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
===== RESTART: E:\KULIAH\SEMESTER 4\prak algostruk\MODUL 8\latihan.py =====
masukkan bilangan positif (<0 untuk mengakhiri): 7
masukkan bilangan positif (<0 untuk mengakhiri): 13
masukkan bilangan positif (<0 untuk mengakhiri): 45
masukkan bilangan positif (<0 untuk mengakhiri): 19
masukkan bilangan positif (<0 untuk mengakhiri): 28
masukkan bilangan positif (<0 untuk mengakhiri): -4
28
19
45
13
7
>>> I

*latihan.py - E:\KULIAH\SEMESTER 4\prak algostruk\MODUL 8\latihan.py...
File Edit Format Run Options Window Help
#features dan properties sebuah stack
PROMPT = "masukkan bilangan positif (<0 untuk mengakhiri): "
myStack = Stack()
value = int(input(PROMPT))
while value >= 0:
    myStack.push(value)
    value = int(input(PROMPT))
while not myStack.isEmpty():
    value = myStack.pop()
    print(value)
```

#### 2. Implementasi stack

##### a. Menggunakan list

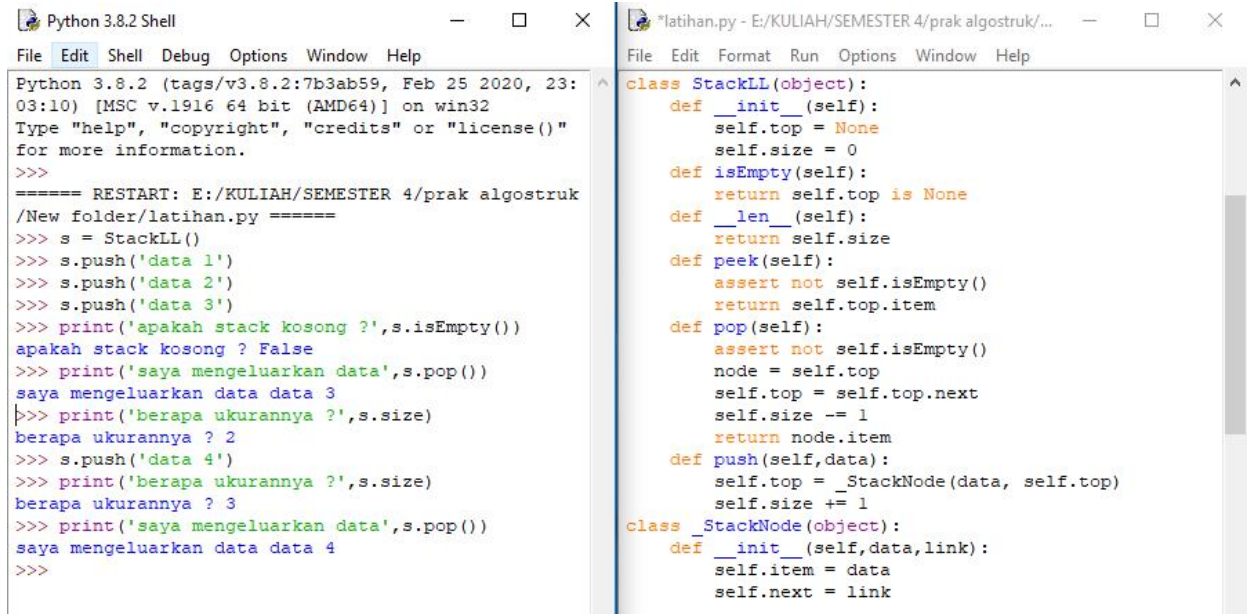


The image shows two side-by-side windows. The left window is a Python 3.8.2 Shell with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). It displays the output of a script where a stack is implemented using a list. The user creates a Stack object, pushes 7, 13, and 45, and then prints the stack's contents and the top element. The right window shows the source code of the script, titled '\*latihan.py - E:\KULIAH\SEMESTER 4\prak algostruk\...'. The code defines a Stack class using a list to store items.

```
===== RESTART: E:\KULIAH\SEMESTER 4\prak algostruk\...
>>> s = Stack()
>>> s.push(7)
>>> s.push(13)
>>> s.push(45)
>>> print('apakah stack kosong?', s.isEmpty())
apakah stack kosong ? False
>>> print('apakah isinya?', s.items)
apakah isinya ? [7, 13, 45]
>>> print('saya mengeluarkan data : ', s.pop())
saya mengeluarkan data : 45
>>> print('nilai yang paling atas : ', s.peek())
nilai yang paling atas : 13
>>> print('apakah isinya?', s.items)
apakah isinya ? [7, 13]
>>>

*latihan.py - E:\KULIAH\SEMESTER 4\prak algostruk\...
File Edit Format Run Options Window Help
#menggunakan list
class Stack(object):
    def __init__(self):
        self.items = []
    def isEmpty(self):
        return len(self) == 0
    def __len__(self):
        return len(self.items)
    def peek(self):
        assert not self.isEmpty()
        return self.items[-1]
    def pop(self):
        assert not self.isEmpty()
        return self.items.pop()
    def push(self, data):
        self.items.append(data)
```

### b. Menggunakan linked list



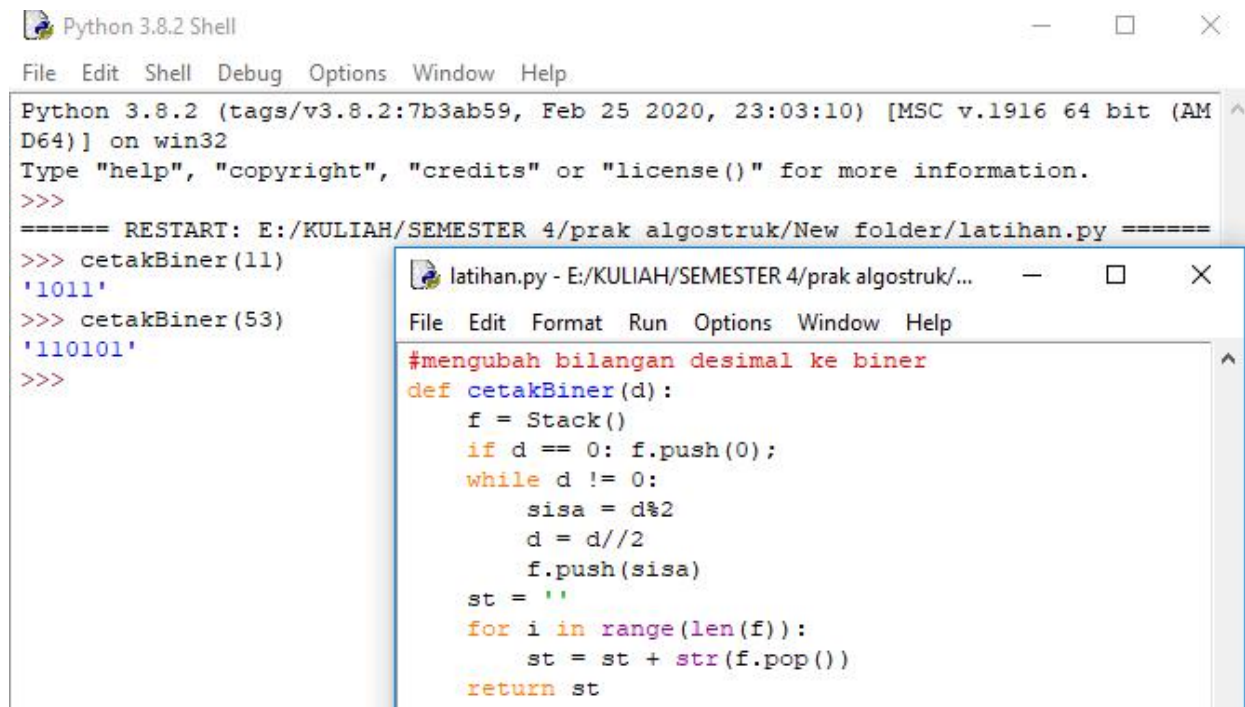
The image shows two windows from a Python 3.8.2 environment. The left window is a Python Shell with the following code and output:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/New folder/latihan.py =====
>>> s = StackLL()
>>> s.push('data 1')
>>> s.push('data 2')
>>> s.push('data 3')
>>> print('apakah stack kosong?', s.isEmpty())
apakah stack kosong? False
>>> print('saya mengeluarkan data', s.pop())
saya mengeluarkan data data 3
>>> print('berapa ukurannya?', s.size)
berapa ukurannya? 2
>>> s.push('data 4')
>>> print('berapa ukurannya?', s.size)
berapa ukurannya? 3
>>> print('saya mengeluarkan data', s.pop())
saya mengeluarkan data data 4
>>>
```

The right window shows the implementation of the StackLL class in latihan.py:

```
class StackLL(object):
    def __init__(self):
        self.top = None
        self.size = 0
    def isEmpty(self):
        return self.top is None
    def __len__(self):
        return self.size
    def peek(self):
        assert not self.isEmpty()
        return self.top.item
    def pop(self):
        assert not self.isEmpty()
        node = self.top
        self.top = self.top.next
        self.size -= 1
        return node.item
    def push(self, data):
        self.top = _StackNode(data, self.top)
        self.size += 1
class _StackNode(object):
    def __init__(self, data, link):
        self.item = data
        self.next = link
```

### 3. Mengubah bilangan decimal ke biner



The image shows two windows from a Python 3.8.2 environment. The left window is a Python Shell with the following code and output:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/New folder/latihan.py =====
>>> cetakBiner(11)
'1011'
>>> cetakBiner(53)
'110101'
>>>
```

The right window shows the implementation of the cetakBiner function in latihan.py:

```
#mengubah bilangan decimal ke biner
def cetakBiner(d):
    f = Stack()
    if d == 0: f.push(0);
    while d != 0:
        sisa = d%2
        d = d//2
        f.push(sisa)
    st = ''
    for i in range(len(f)):
        st = st + str(f.pop())
    return st
```