Nama : Rifqi Alwan

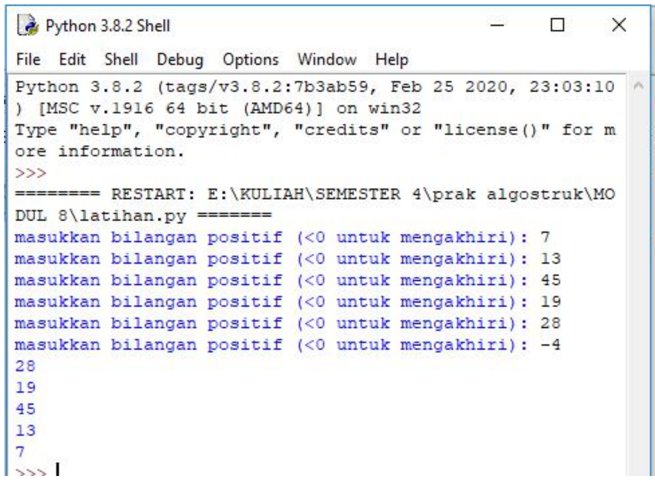
NIM : L200180010

Kelas A

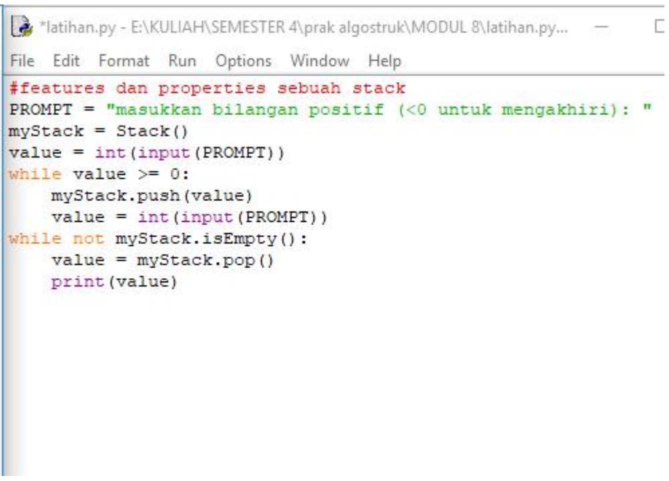
LATIHAN MODUL 8

QUEUE

1. Features dan properties sebuah stack

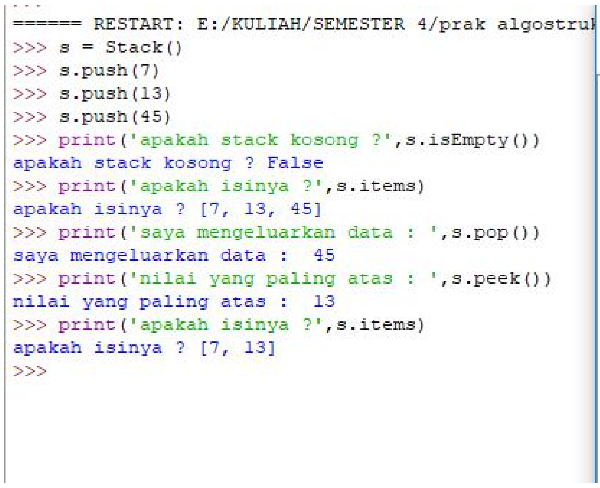


```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10
) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for m
ore information.
>>>
======== RESTART: E:\KULIAH\SEMESTER 4\prak algostruk\MO
DUL 8\latihan.py =======
masukkan bilangan positif (<0 untuk mengakhiri):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

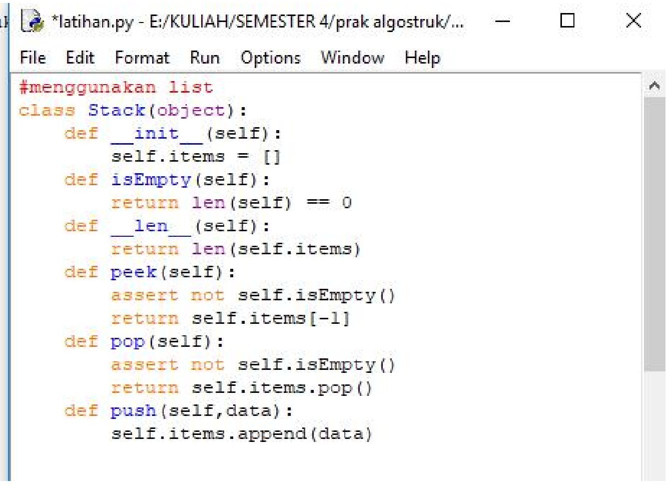

*latihan.py - E:\KULIAH\SEMESTER 4\prak algostruk\MODUL 8\latihan.py...
File Edit Format Run Options Window Help

```
#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

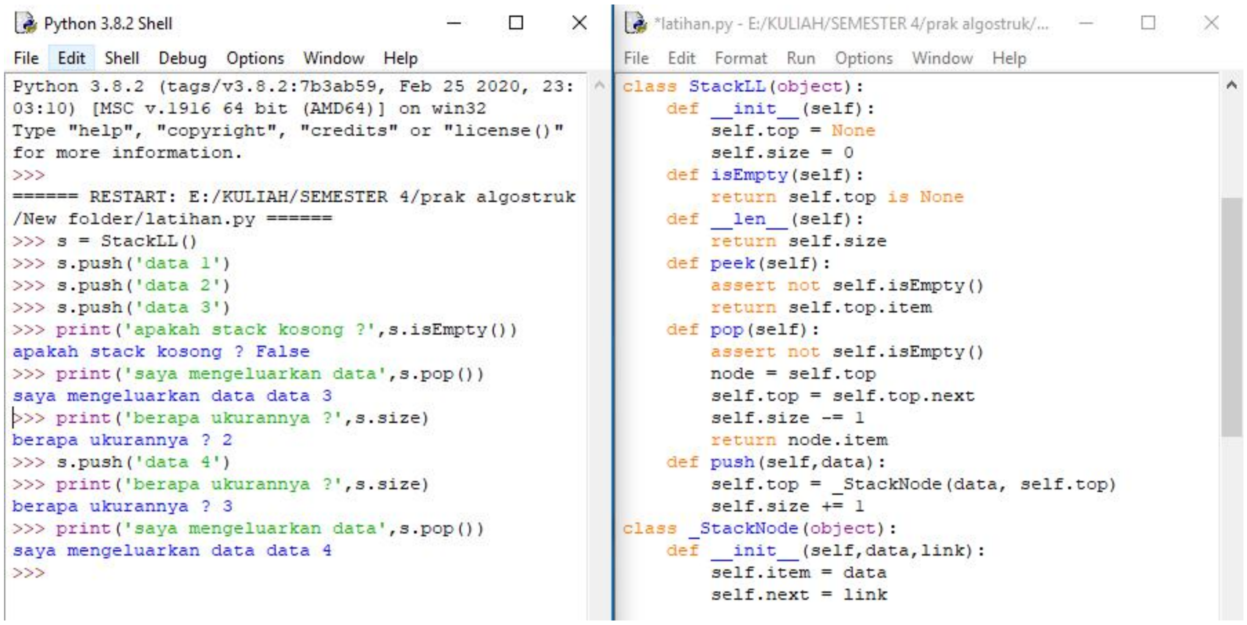


```
====== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk
>>> s = Stack()
>>> s.push(7)
>>> s.push(13)
>>> s.push(45)
>>> print('apakah stack kosong ?',s.isEmpty())
apakah stack kosong ? False
>>> print('apakah isinya ?',s.items)
apakah isinya ? [7, 13, 45]
>>> print('saya mengeluarkan data : ',s.pop())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

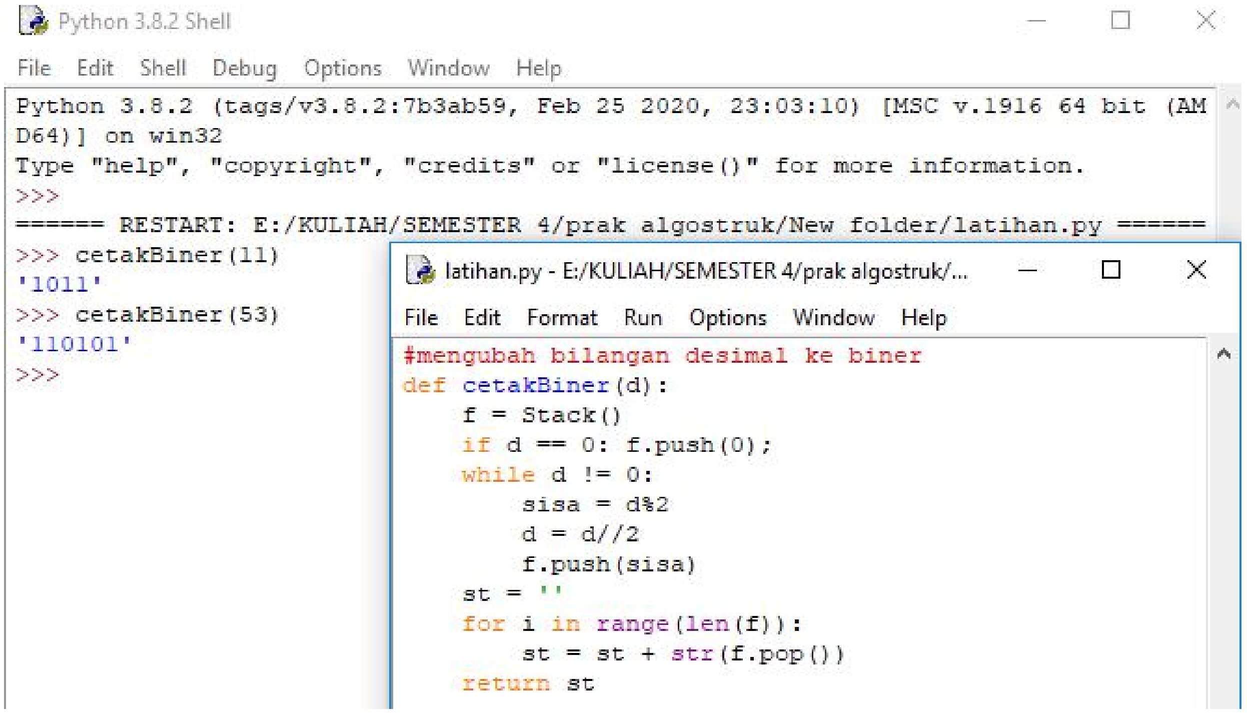
b. Menggunakan linked list


Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
*latihan.py - E:/KULIAH/SEMESTER 4/prak algostruk/...
File Edit Format Run Options Window Help

```
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
>>>
```

```
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
```

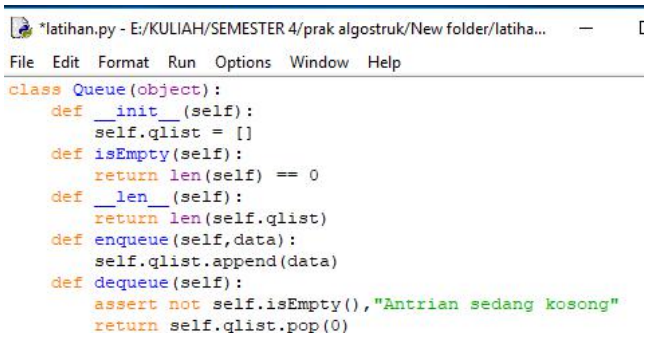
3. Mengubah bilangan decimal ke biner


Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
latihan.py - E:/KULIAH/SEMESTER 4/prak algostruk/...
File Edit Format Run Options Window Help

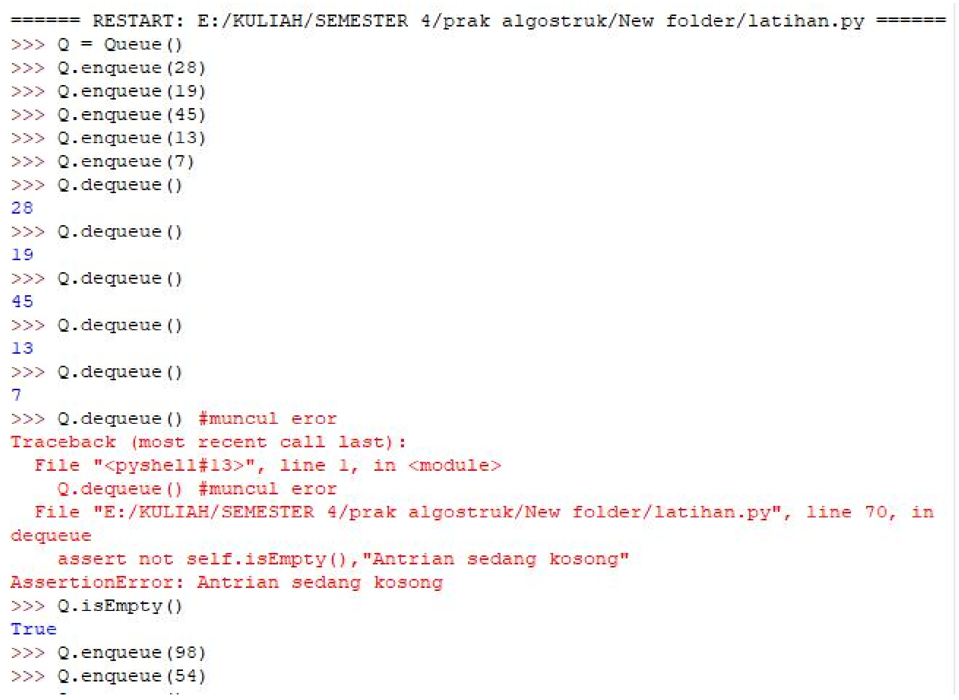
```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/New folder/latihan.py ======
>>> cetakBiner(11)
'1011'
>>> cetakBiner(53)
'110101'
>>>
```

```
#mengubah bilangan desimal 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
```

4. Implementasi queue

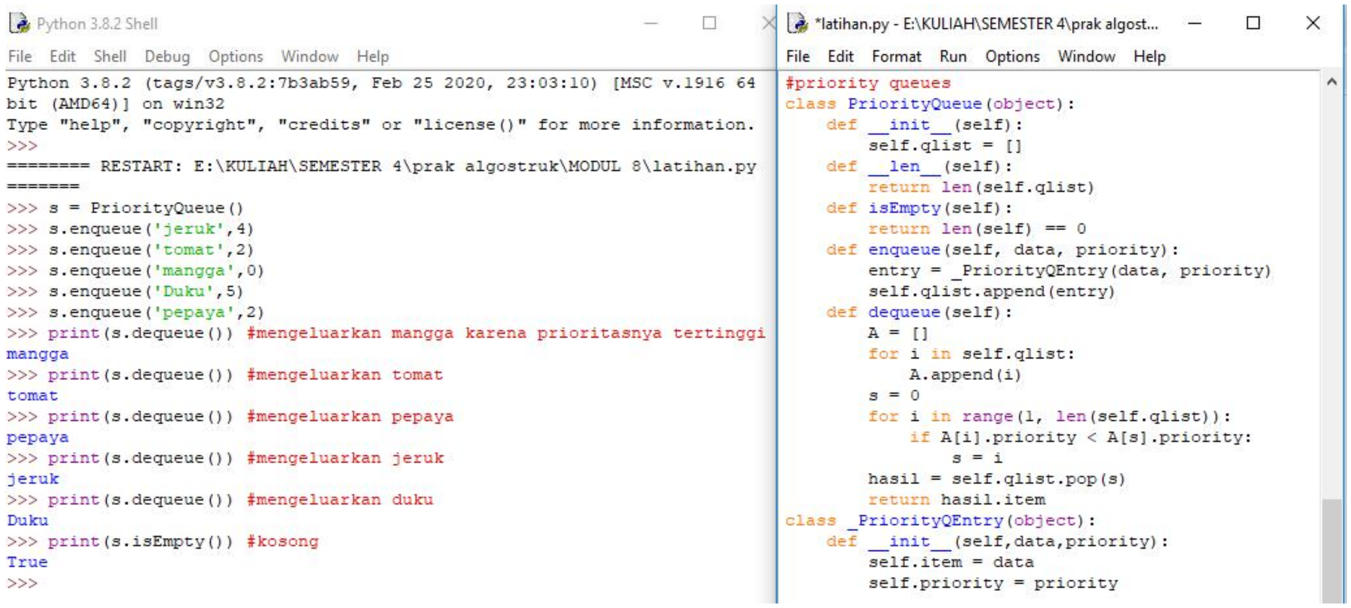

*latihan.py - E:/KULIAH/SEMESTER 4/prak algostruk/New folder/latiha...
File Edit Format Run Options Window Help

```
class Queue(object):
    def __init__(self):
        self.qlist = []
    def isEmpty(self):
        return len(self) == 0
    def __len__(self):
        return len(self.qlist)
    def enqueue(self,data):
        self.qlist.append(data)
    def dequeue(self):
        assert not self.isEmpty(),"Antrian sedang kosong"
        return self.qlist.pop(0)
```



```
====== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/New folder/latihan.py ======
>>> Q = Queue()
>>> Q.enqueue(28)
>>> Q.enqueue(19)
>>> Q.enqueue(45)
>>> Q.enqueue(13)
>>> Q.enqueue(7)
>>> Q.dequeue()
28
>>> Q.dequeue()
19
>>> Q.dequeue()
45
>>> Q.dequeue()
13
>>> Q.dequeue()
7
>>> Q.dequeue() #muncul eror
Traceback (most recent call last):
  File "<pyshell#13>", line 1, in <module>
    Q.dequeue() #muncul eror
  File "E:/KULIAH/SEMESTER 4/prak algostruk/New folder/latihan.py", line 70, in
dequeue
    assert not self.isEmpty(),"Antrian sedang kosong"
AssertionError: Antrian sedang kosong
>>> Q.isEmpty()
True
>>> Q.enqueue(98)
>>> Q.enqueue(54)
```

5. Priority queues


Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
*latihan.py - E:\KULIAH\SEMESTER 4\prak algost...
File Edit Format Run Options Window Help

```
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\MODUL 8\latihan.py
=======
>>> s = PriorityQueue()
>>> s.enqueue('jeruk',4)
>>> s.enqueue('tomat',2)
>>> s.enqueue('mangga',0)
>>> s.enqueue('Duku',5)
>>> s.enqueue('pepaya',2)
>>> print(s.dequeue()) #mengeluarkan mangga karena prioritasnya tertinggi
mangga
>>> print(s.dequeue()) #mengeluarkan tomat
tomat
>>> print(s.dequeue()) #mengeluarkan pepaya
pepaya
>>> print(s.dequeue()) #mengeluarkan jeruk
jeruk
>>> print(s.dequeue()) #mengeluarkan duku
Duku
>>> print(s.isEmpty()) #kosong
True
>>> 
```

```
#priority queues
class PriorityQueue(object):
    def __init__(self):
        self.qlist = []
    def __len__(self):
        return len(self.qlist)
    def isEmpty(self):
        return len(self) == 0
    def enqueue(self, data, priority):
        entry = _PriorityQEntry(data, priority)
        self.qlist.append(entry)
    def dequeue(self):
        A = []
        for i in self.qlist:
            A.append(i)
        s = 0
        for i in range(1, len(self.qlist)):
            if A[i].priority < A[s].priority:
                s = i
        hasil = self.qlist.pop(s)
        return hasil.item
class _PriorityQEntry(object):
    def __init__(self,data,priority):
        self.item = data
        self.priority = priority
```