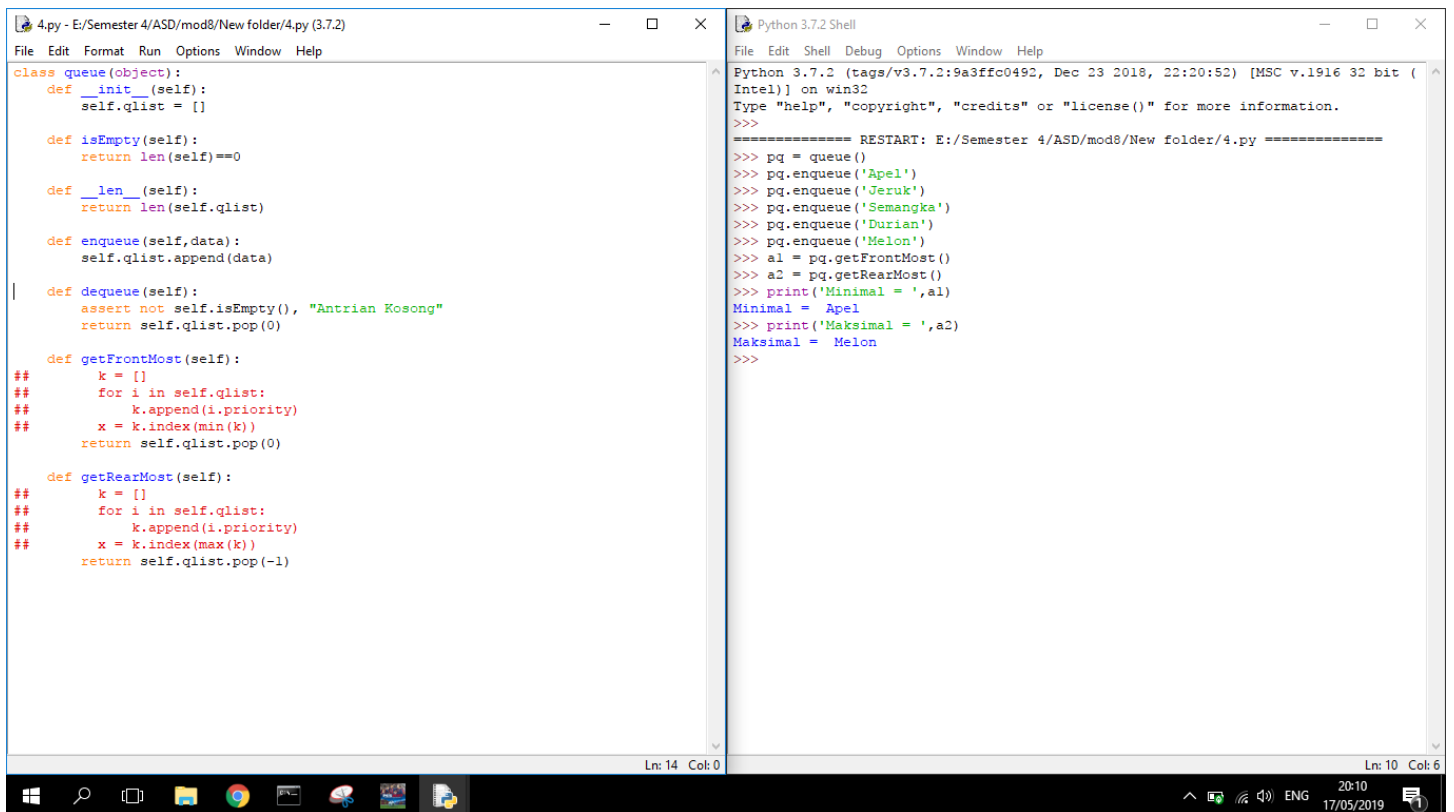


Nama : Alvian Harisnur  
NIM : L200170132  
Kelas : D

## Modul 8

Soal – soal untuk mahasiswa

4.



The screenshot shows a Python IDE with two windows. The left window, titled '4.py - E:/Semester 4/ASD/mod8/New folder/4.py (3.7.2)', contains the following code:

```
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 Kosong"
        return self.qlist.pop(0)

    def getFrontMost(self):
        k = []
        for i in self.qlist:
            k.append(i.priority)
        x = k.index(min(k))
        return self.qlist.pop(0)

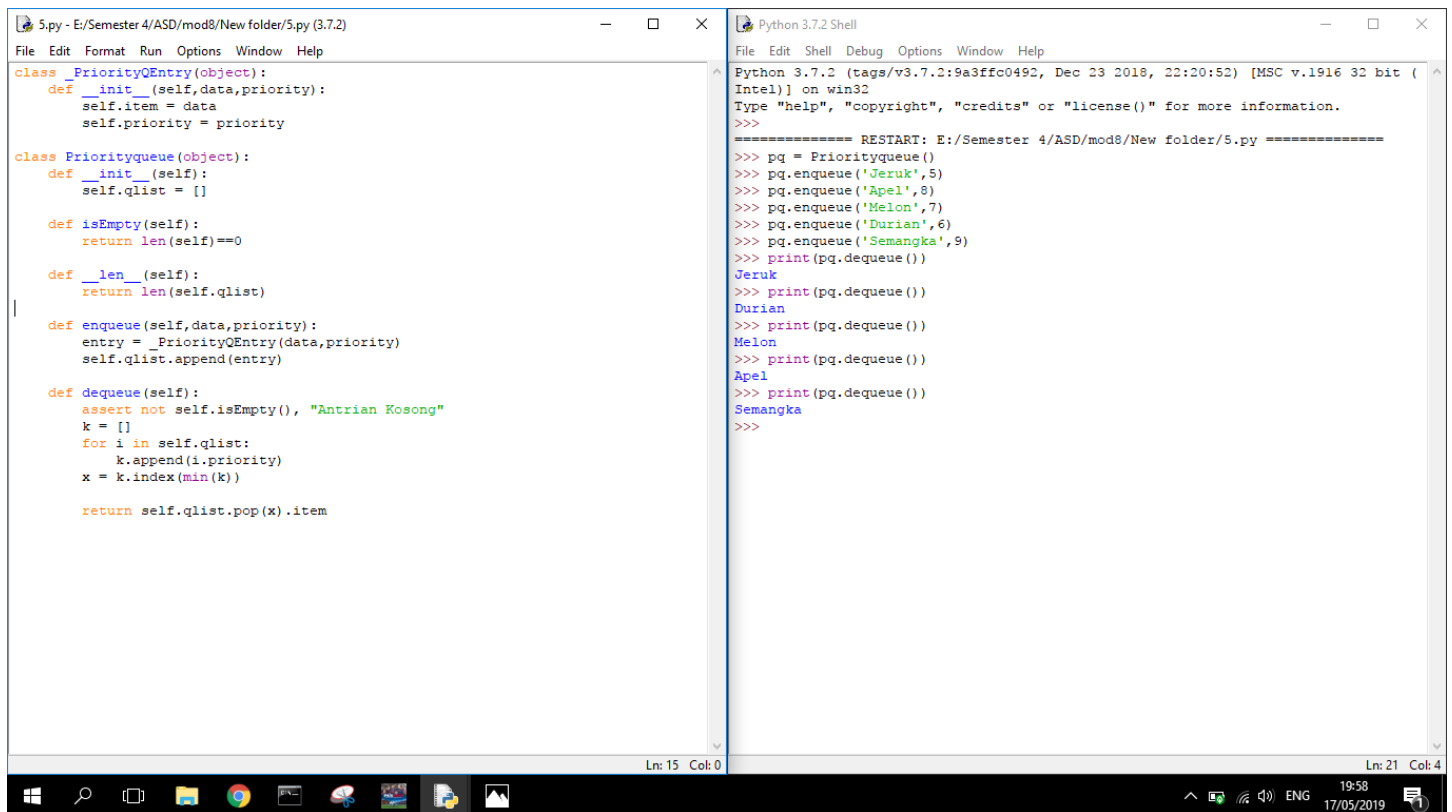
    def getRearMost(self):
        k = []
        for i in self.qlist:
            k.append(i.priority)
        x = k.index(max(k))
        return self.qlist.pop(-1)
```

The right window, titled 'Python 3.7.2 Shell', shows the execution of the code:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/Semester 4/ASD/mod8/New folder/4.py =====
>>> pq = queue()
>>> pq.enqueue('Apel')
>>> pq.enqueue('Jeruk')
>>> pq.enqueue('Semangka')
>>> pq.enqueue('Durian')
>>> pq.enqueue('Melon')
>>> a1 = pq.getFrontMost()
>>> a2 = pq.getRearMost()
>>> print('Minimal = ',a1)
Minimal =  Apel
>>> print('Maksimal = ',a2)
Maksimal =  Melon
>>>
```

The taskbar at the bottom shows the Windows Start button, search icon, task view icon, and several application icons. The system clock in the bottom right corner displays '20:10' and '17/05/2019'.

5.



The image shows a screenshot of a Python IDE with two windows. The left window, titled '5.py - E:/Semester 4/ASD/mod8/New folder/5.py (3.7.2)', contains the implementation of a Priority Queue. The right window, titled 'Python 3.7.2 Shell', shows the execution of the code, including a restart message and the output of the dequeue operations.

```
class _PriorityQEntry(object):
    def __init__(self, data, priority):
        self.item = data
        self.priority = priority

class PriorityQueue(object):
    def __init__(self):
        self.qlist = []

    def isEmpty(self):
        return len(self)==0

    def __len__(self):
        return len(self.qlist)

    def enqueue(self, data, priority):
        entry = _PriorityQEntry(data, priority)
        self.qlist.append(entry)

    def dequeue(self):
        assert not self.isEmpty(), "Antrian Kosong"
        k = []
        for i in self.qlist:
            k.append(i.priority)
        x = k.index(min(k))
        return self.qlist.pop(x).item

>>> pq = PriorityQueue()
>>> pq.enqueue('Jeruk', 5)
>>> pq.enqueue('Apel', 8)
>>> pq.enqueue('Melon', 7)
>>> pq.enqueue('Durian', 6)
>>> pq.enqueue('Semangka', 9)
>>> print(pq.dequeue())
Jeruk
>>> print(pq.dequeue())
Durian
>>> print(pq.dequeue())
Melon
>>> print(pq.dequeue())
Apel
>>> print(pq.dequeue())
Semangka
>>>
```

Ln: 15 Col: 0

Ln: 21 Col: 4

19:58  
17/05/2019