

# LAPORAN PRAKTIKUM ALGORITMA STRUKTUR DATA

## MODUL 8

### “STACK AND QUEUE”

Nama : Aiza Fravy Qanza

NIM : L200170144

Kelas : D

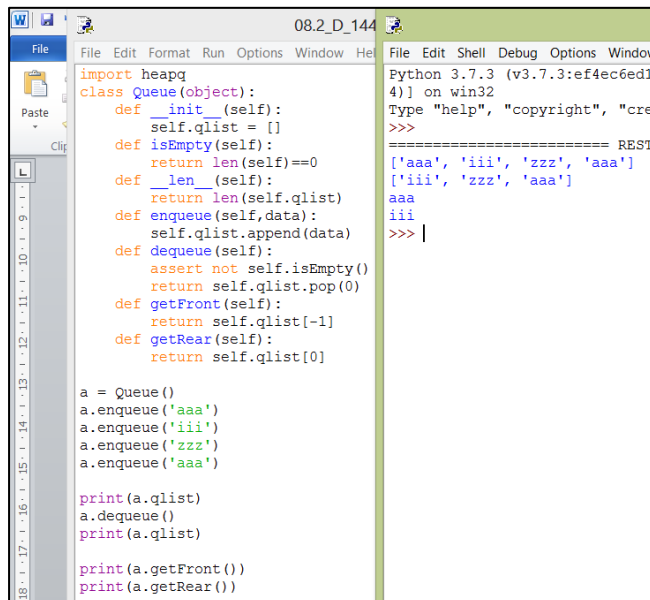
#### Soal-soal untuk Mahasiswa

>> Queue

4. Tulis dua metode berikut ke class Queue dan class PriorityQueue diatas

Metode untuk mengetahui item yang paling depan tanpa menghapusnya dan Metode untuk mengetahui item yang paling belakang tanpa menghapusnya

- Queue



```
import heapq

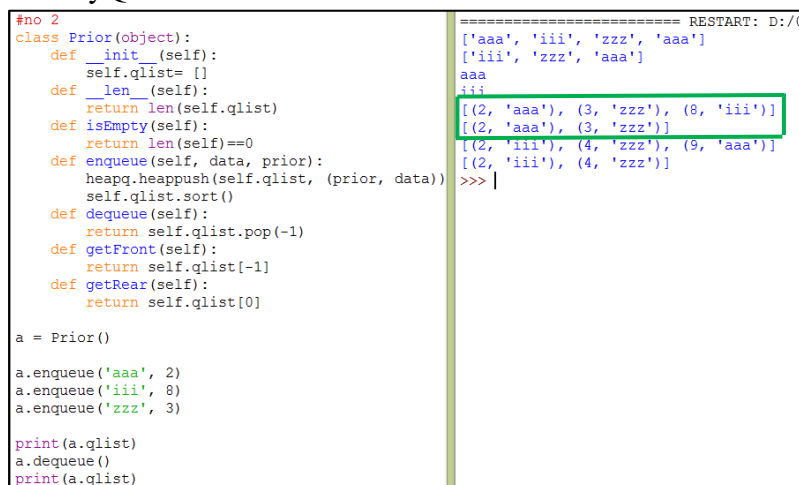
class Queue(object):
    def __init__(self):
        self.qlist = []
    def isEmpty(self):
        return len(self)==0
    def __len__(self):
        return len(self.qlist)
    def enqueue(self, data):
        self.qlist.append(data)
    def dequeue(self):
        assert not self.isEmpty()
        return self.qlist.pop(0)
    def getFront(self):
        return self.qlist[-1]
    def getRear(self):
        return self.qlist[0]

a = Queue()
a.enqueue('aaa')
a.enqueue('iii')
a.enqueue('zzz')
a.enqueue('aaa')

print(a.qlist)
a.dequeue()
print(a.qlist)

print(a.getFront())
print(a.getRear())
```

- PriorityQueue



```
#no 2
class Prior(object):
    def __init__(self):
        self.qlist= []
    def __len__(self):
        return len(self.qlist)
    def isEmpty(self):
        return len(self)==0
    def enqueue(self, data, prior):
        heapq.heappush(self.qlist, (prior, data))
        self.qlist.sort()
    def dequeue(self):
        return self.qlist.pop(-1)
    def getFront(self):
        return self.qlist[-1]
    def getRear(self):
        return self.qlist[0]

a = Prior()

a.enqueue('aaa', 2)
a.enqueue('iii', 8)
a.enqueue('zzz', 3)

print(a.qlist)
a.dequeue()
print(a.qlist)
```

5. Pada class PriorityQueue diatas, metode dequeue() belum diimplementasikan. Tulislah metode dequeue() ini dengan memperhatikan syarat-syarat seperti yang telah dicantumkan di halaman 81

```
print(a.dequeue())

#no 3
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, prior):
        heapq.heappush(self.qlist, (prior, data))
        self.qlist.sort()
    def dequeue(self):
        return self.qlist.pop(-1)

a = PriorityQueue()

a.enqueue('aaa', 9)
a.enqueue('iii', 2)
a.enqueue('zzz', 4)

print(a.qlist)
a.dequeue()
print(a.qlist)
```

```
===== RESTART: D:/08
['aaa', 'iii', 'zzz', 'aaa']
['iii', 'zzz', 'aaa']
aaa
iii
[(2, 'aaa'), (3, 'zzz'), (8, 'iii')]
[(2, 'aaa'), (3, 'zzz')]
[(2, 'iii'), (4, 'zzz'), (9, 'aaa')]
[(2, 'iii'), (4, 'zzz')]
>>>
```