

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

**MODUL 8**

**STACK AND QUEUES**



**Oleh : Fariz Taufiqul Hafidz**

**Nim : L200210192**

**Kelas : F**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2023**

## A. Queue

4. Tulis dua metode berikut ke class Queue dan class PriorityQueue di atas

- Metode untuk mengetahui item yang paling depan tanpa menghapusnya  
def getFrontMost(self):

## Tulis perintahnya di sini

- Metode untuk mengetahui item yang paling belakang tanpa menghapusnya  
def getRearMost(self):

## Tulis perintahnya di sini

```
1 ##### Nomor 4 #####
2 class Queue():
3     def __init__(self):
4         self.qlist = []
5     def is_empty(self):
6         return len(self) == 0
7     def __len__(self):
8         return len(self.qlist)
9     def enqueue(self, data):
10        self.qlist.append(data)
11    def dequeue(self):
12        assert not self.is_empty(), 'Antrian sedang kosoong'
13        return self.qlist.pop(0)
14    def get_front_most(self):
15        assert not self.is_empty(), 'Antrian sedang kosoong'
16        return self.qlist[0]
17    def get_rear_most(self):
18        assert not self.is_empty(), 'Antrian sedang kosoong'
19        return self.qlist[-1]
20
21    Y = Queue()
22    Y.enqueue(43)
23    Y.enqueue(21)
24    Y.enqueue(7)
25    Y.enqueue(38)
26    Y.enqueue(16)
27
28    print(Y.get_front_most())
29    print(Y.get_rear_most())
30    print(Y.qlist)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS E:\Kuliah\Semester 4\Praktikum ASD\Modul 8> & 'C:\Users\Asus\AppData\Local\Programs\Python\Python39-2023.6.1\pythonFiles\lib\python\debugpy\adapter\..\..\debu
43
16
[43, 21, 7, 38, 16]
```

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

```
1 ##### Nomor 5 #####
2 class PriorityQueue():
3     def __init__(self):
4         self.qlist = []
5     def __len__(self):
6         return len(self.qlist)
7     def is_empty(self):
8         return len(self) == 0
9     def enqueue(self, data, priority):
10        entry = _PriorityQEntry(data, priority)
11        self.qlist.append(entry)
12    def get_front_most(self):
13        assert not self.is_empty(), 'Antrian sedang kosoong'
14        return self.qlist[0]
15    def get_rear_most(self):
16        assert not self.is_empty(), 'Antrian sedang kosoong'
17        return self.qlist[-1]
18    def dequeue(self):
19        A = []
20        for i in self.qlist:
21            A.append(i)
22        s = 0
23        for i in range(1, len(self.qlist)):
24            if A[i].priority < A[s].priority:
25                s = i
26        hasil = self.qlist.pop(s)
27        return hasil.item
28
29 class _PriorityQEntry():
30     def __init__(self, data, priority):
31         self.item = data
32         self.priority = priority
33
34 f = PriorityQueue()
35 f.enqueue("Predator",3)
36 f.enqueue("Omen",5)
37 f.enqueue("ROG",1)
38 f.enqueue("Legion",2)
39 f.enqueue("Victus",4)
40
41 print(f.dequeue())
42 print(f.dequeue())
43 print(f.dequeue())
44 print(f.dequeue())
45 print(f.dequeue())
```

```
ROG
Legion
Predator
Victus
Omen
PS E:\Kuliah\Semester 4\Praktikum ASD\Modul 8>
```