

**PRAKTIKUM ALGORITMA STRUKTUR DATA**

**MODUL 7**

*Stacks and Queues*



**Disusun oleh:**

**Adinda Aulia Hapsari**

**L200220037**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH SURAKARTA  
TAHUN 2024**

## 8.8 Soal-soal untuk Mahasiswa

### Stacks

1. Buatlah program untuk mengubah representasi suatu bilangan dari basis sepuluh kebasis dua. Berikut ini contoh pemanggilannya.

```
1 >>> cetakHexa(12)
2 'C'
3 >>> cetakHexa(31)
4 '1F'
5 >>> cetakHexa(229)
6 'E5'
7 >>> cetakHexa(255)
8 'FF'
9 >>> cetakHexa(31519)
10 '7B1F'
```

Perhatikan bahwa sisa pembagian tidak hanya 0 dan 1, namun bisa 0 sampai 9 dan bahkan 10, 11, 12, 13, 14, 15. Kamu harus memetakan angka-angka yang lebih dari 9 kelambang A, B, C, D, E, dan F.

1.py	2.py	3.py	5.py	Priority.py	Queue.py
------	------	------	------	-------------	----------

```
modul8 > 1.py > ...
1 print('\n--- Oleh L200220037 ---')
2 class Stack(object):
3     def __init__(self):
4         self.items = []
5     def isEmpty(self):
6         return len(self)==0
7     def __len__(self):
8         return len(self.items)
9     def peek(self):
10        assert not self.isEmpty()
11        return self.items[-1]
12    def pop(self):
13        assert not self.isEmpty()
14        return self.items.pop()
15    def push(self,data):
16        self.items.append(data)
17
18    def cetakHexa(d):
19        f = Stack()
20        bil_hexa = ['A','B','C','D','E','F']
21        if d==0: f.push(0);
22        while d!=0:
23            sisa = d%16
24            if (sisa>9):
25                sisa = sisa-10
26                sisa = bil_hexa[sisa]
27
28            d=d//16
29            f.push(sisa)
30        st = ""
31        for i in range(len(f)):
32            st = st+str(f.pop())
33        return st
34
35 print(cetakHexa(12))
36 print(cetakHexa(31))
37 print(cetakHexa(229))
38 print(cetakHexa(225))
39 print(cetakHexa(31519))
40
41
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA> & C
4/PRAKTIKUM ALGORITMA STRUKTUR DATA/modul8/1.py"

--- Oleh L200220037 ---
C
1F
E5
E1
7B1F
PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA>
```

2. Eksekusi program berikut dengan pensil dan kertas, dan tunjukkan isi stack-nya pada setiap langkah.

```
1 nilai = Stack()
2 for i in range(16 )
3 if i % 3 == 0:
4 nilai.push( i )
```

```
1.py 2.py x 3.py 5.py Priority.py Queue.py
modul8 > 2.py > ...
1 print('\n--- Oleh L200220037 ---')
2 class Stack(object):
3     def __init__(self):
4         self.items = []
5     def isEmpty(self):
6         return len(self)==0
7     def __len__(self):
8         return len(self.items)
9     def peek(self):
10        assert not self.isEmpty()
11        return self.items[-1]
12    def pop(self):
13        assert not self.isEmpty()
14        return self.items.pop()
15    def push(self,data):
16        self.items.append(data)
17
18 nilai =Stack()
19 for i in range(16):
20     if i%3==0:
21         nilai.push(i)
22 print(nilai.items)
23
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA> &
4/PRAKTIKUM ALGORITMA STRUKTUR DATA/modul8/2.py"

--- Oleh L200220037 ---
[0, 3, 6, 9, 12, 15]
PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA>
```

3. Eksekusi program berikut dengan pensil dan kertas, dan tunjukkan isi stack-nya pada setiap langkah.

```
1 nilai = Stack()
```

```

2 for i in range( 16 ) :
3 if i % 3 == 0 :
4 nilai.push( i )
5 elif i % 4 == 0 :
6 nilai.pop()

```

The screenshot shows a Python IDE with a file explorer at the top containing files: 1.py, 2.py, 3.py (selected), 5.py, Priority.py, and Queue.py. The main editor displays the following Python code:

```

1 print('\n--- Oleh L200220037 ---')
2 class Stack(object):
3     def __init__(self):
4         self.items = []
5     def isEmpty(self):
6         return len(self)==0
7     def __len__(self):
8         return len(self.items)
9     def peek(self):
10        assert not self.isEmpty()
11        return self.items[-1]
12    def pop(self):
13        assert not self.isEmpty()
14        return self.items.pop()
15    def push(self,data):
16        self.items.append(data)
17 nilai = Stack()
18 for i in range(16):
19     if i%3==0:
20         nilai.push(i)
21     elif i%4==0:
22         nilai.pop()
23 print(nilai.items)
24

```

Below the editor is a terminal window with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, **TERMINAL**, and PORTS. The terminal output is as follows:

```

PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA> &
4/PRAKTIKUM ALGORITMA STRUKTUR DATA/modul8/3.py"

--- Oleh L200220037 ---
[0, 9, 12, 15]
PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA>

```

## Queues

4. Tulis dua metode berikut keclass Queue dan class Priority Queue diatas.
  - 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.py 2.py 3.py 5.py Priority.py X Queue.py
modul8 > Priority.py > ...
1 print('\n--- Oleh L200220037 ---')
2 import heapq
3
4 class PriorityQueue(object):
5     def __init__(self):
6         self.qlist = []
7
8     def __len__(self):
9         return len(self.qlist)
10
11     def isEmpty(self):
12         return len(self.qlist) == 0
13
14     def enqueue(self, data, prior):
15         heapq.heappush(self.qlist, (prior, data))
16         self.qlist.sort()
17
18     def dequeue(self):
19         return self.qlist.pop(-1)
20
21     def getFrontMost(self):
22         return self.qlist[-1]
23
24     def getRearMost(self):
25         return self.qlist[0]
26
27 A = PriorityQueue()
28 A.enqueue("Jeruk", 4)
29 A.enqueue("Tomat", 2)
30 A.enqueue("Mangga", 0)
31 A.enqueue("Duku", 5)
32 A.enqueue("Pepaya", 2)
33
34 print(A.qlist)
35 A.dequeue()
36 print(A.qlist)
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS
PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA> & C:/Users/.../4/PRAKTIKUM ALGORITMA STRUKTUR DATA/modul8/Priority.py"				
--- Oleh L200220037 ---				
[(0, 'Mangga'), (2, 'Pepaya'), (2, 'Tomat'), (4, 'Jeruk'), (5, 'Duku')]				
[(0, 'Mangga'), (2, 'Pepaya'), (2, 'Tomat'), (4, 'Jeruk')]				
PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA> █				

```
1.py 2.py 3.py 5.py Priority.py Queue.py X
modul8 > Queue.py > ...
1 print('\n--- Oleh L200220037 ---')
2 class Queue(object):
3     def __init__(self):
4         self.qlist = []
5
6     def isEmpty(self):
7         return len(self.qlist) == 0
8
9     def __len__(self):
10        return len(self.qlist)
11
12    def enqueue(self, data):
13        self.qlist.append(data)
14
15    def dequeue(self):
16        assert not self.isEmpty()
17        return self.qlist.pop(0)
18
19    def getFrontMost(self):
20        return self.qlist.pop(0)
21
22    def getRearMost(self):
23        return self.qlist.pop(-1)
24
25 A = Queue()
26 A.enqueue("Jeruk")
27 A.enqueue("Tomat")
28 A.enqueue("Mangga")
29 A.enqueue("Duku")
30 A.enqueue("Pepaya")
31 A.enqueue("Kelengkeng")
32
33 front = A.getFrontMost()
34 rear = A.getRearMost()
35 print("Nilai maksimal = ", front)
36 print("Nilai minimal = ", rear)
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS
<pre> PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA&gt; 4/PRAKTIKUM ALGORITMA STRUKTUR DATA/modul8/Queue.py"  --- Oleh L200220037 --- Nilai maksimal = Jeruk Nilai minimal = Kelengkeng PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA&gt; </pre>				

5. Pada class Priority Queue diatas, metode dequeue() belum diimplementasikan. Tulislah metode dequeue() ini dengan memperhatikan syarat-syarat seperti yang telah dicantumkan dihalaman81.

1.py	2.py	3.py	5.py	Priority.py	Queue.py										
<pre> modul8 &gt; 5.py &gt; ... 1 print('\n--- Oleh L200220037 ---') 2 class _PriorityQEntry(object): 3     def __init__(self, data, priority): 4         self.item = data 5         self.priority = priority 6 7 class PriorityQueue(object): 8     def __init__(self): 9         self.qlist = [] 10 11     def __len__(self): 12         return len(self.qlist) 13 14     def isEmpty(self): 15         return len(self.qlist) == 0 16 17     def enqueue(self, data, priority): 18         entry = _PriorityQEntry(data, priority) 19         self.qlist.append(entry) 20 21     def dequeue(self): 22         assert not self.isEmpty() 23         k = [] 24         for i in self.qlist: 25             k.append(i.priority) 26         x = k.index(min(k)) 27         return self.qlist.pop(x).item 28 29 A = PriorityQueue() 30 A.enqueue("Jeruk", 4) 31 A.enqueue("Tomat", 2) 32 A.enqueue("Mangga", 0) 33 A.enqueue("Duku", 5) 34 A.enqueue("Pepaya", 2) 35 36 print(A.dequeue()) 37 print(A.dequeue()) 38 print(A.dequeue()) 39 print(A.dequeue()) 40 print(A.dequeue()) 41 </pre>															
<table border="1"> <thead> <tr> <th>PROBLEMS</th> <th>OUTPUT</th> <th>DEBUG CONSOLE</th> <th>TERMINAL</th> <th>PORTS</th> </tr> </thead> <tbody> <tr> <td colspan="5"> <pre> PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA&gt; 4/PRAKTIKUM ALGORITMA STRUKTUR DATA/modul8/5.py"  --- Oleh L200220037 --- Mangga Tomat Pepaya Jeruk Duku PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA&gt; </pre> </td> </tr> </tbody> </table>						PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS	<pre> PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA&gt; 4/PRAKTIKUM ALGORITMA STRUKTUR DATA/modul8/5.py"  --- Oleh L200220037 --- Mangga Tomat Pepaya Jeruk Duku PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA&gt; </pre>				
PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS											
<pre> PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA&gt; 4/PRAKTIKUM ALGORITMA STRUKTUR DATA/modul8/5.py"  --- Oleh L200220037 --- Mangga Tomat Pepaya Jeruk Duku PS D:\KULIAH\MATERI\SEMESTER 4\PRAKTIKUM ALGORITMA STRUKTUR DATA&gt; </pre>															