LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
MODUL 8
STACKS AND QUEUES

Disusun oleh:

Bima Triadmaja

L200210137

E

TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

2022/2023

8.8 Soal-soal untuk Mahasiswa

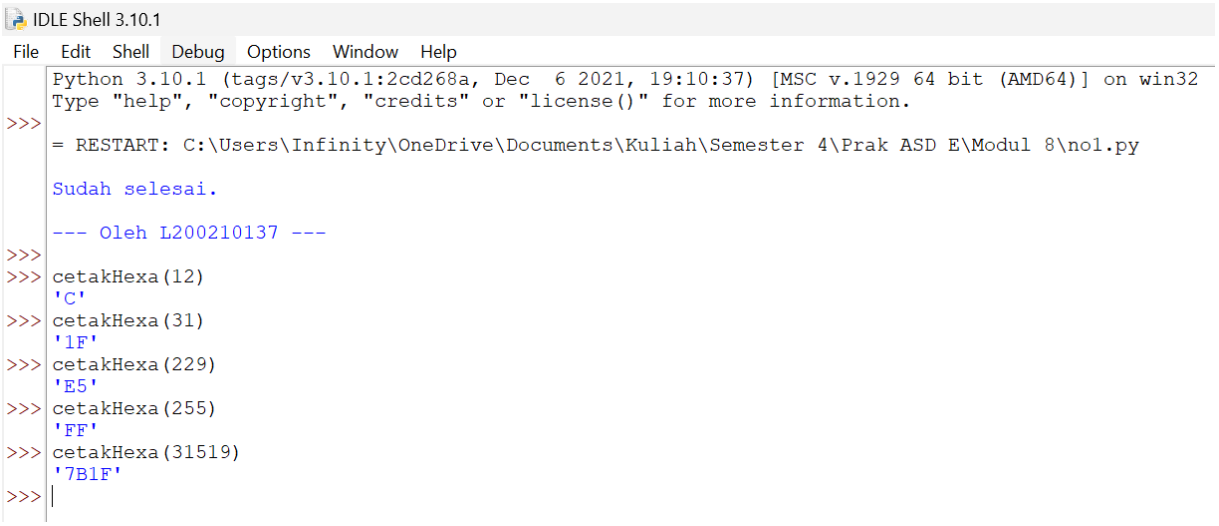
Nomor 1 :

```
#NO.1
class Stack():
    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)

def cetakHexa(d):
    f = Stack()
    if d == 0: f.push(0);
    while d != 0:
        if d%16 == 10:
            sisa = "A"
        elif d%16 == 11:
            sisa = "B"
        elif d%16 == 12:
            sisa = "C"
        elif d%16 == 13:
            sisa = "D"
        elif d%16 == 14:
            sisa = "E"
        elif d%16 == 15:
            sisa = "F"
        else:
            sisa = d%16
        d=d//16
        f.push(sisa)
    st = ""
    for i in range (len(f)):
        st = st + str(f.pop())
    return st
```

```
print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```



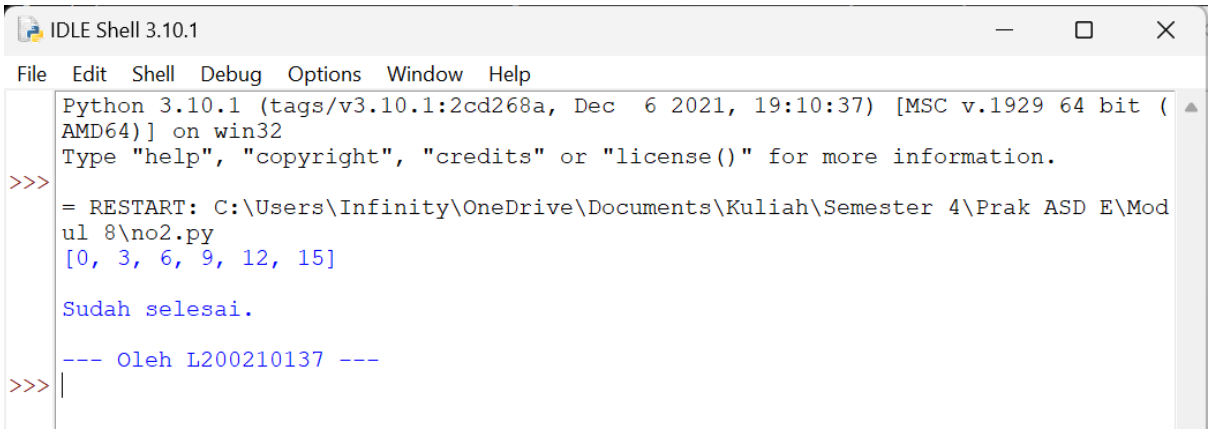
Nomor 2 :

```
#NO.2
class Stack():
    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)

nilai = Stack()
for i in range(16):
    if i%3==0:
        nilai.push(i)
print(nilai.items)
```

```
print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```



Nomor 3 :

```
#NO.3
class Stack():
    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)

nilai = Stack()
for i in range(16):
    if i % 3 == 0:
        nilai.push(i)
    elif i%4==0:
        nilai.pop()
print('')
```

```
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```

```
= RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Mod
ul 8\no3.py

Sudah selesai.

--- Oleh L200210137 ---
>>> nilai.pop()
15
>>> nilai.pop()
12
>>> nilai.pop()
9
>>> nilai.pop()
0
>>> nilai.pop()
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    nilai.pop()
  File "C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul
8\no3.py", line 13, in pop
    assert not self.isEmpty()
AssertionError
>>>
```

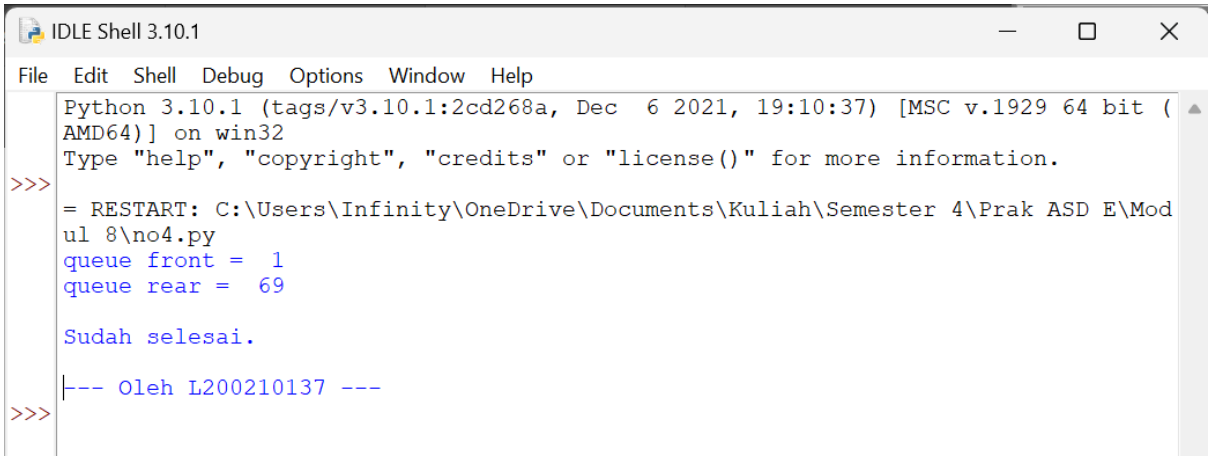
Nomor 4 :

```
#NO.4 Metode 1 (Class Queue)
class Queue():
    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 getFrontMost(self):
        return self.qlist[0]
    def getRearMost(self):
        return self.qlist[len(self.qlist)-1]

q = Queue()
q.enqueue(1)
q.enqueue(12)
q.enqueue(56)
```

```
q.enqueue(69)
print("queue front = ", q.getFrontMost())
print("queue rear = ", q.getRearMost())
print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```




```
#NO.4 Metode 2 (Class Priority Queue)
class PriorityQueue():
    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 getFrontMost(self):
        x = 0
        while self.qlist[x].priority != 0:
            x+=1
        return self.qlist[x].item
    def getRearMost(self):
        thelist = []
        for i in self.qlist:
            thelist.append(i.priority)
        print (self.qlist[thelist.index(max(thelist))].item)

class PriorityQEntry():
```

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

a = PriorityQueue()
a.enqueue("Jeruk", 4)
a.enqueue("Tomat", 2)
a.enqueue("Mangga", 0)
a.enqueue("Duku", 5)
a.enqueue("Pepaya", 2)

print(a.getFrontMost())
print(a.getRearMost())
print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```

```
= RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Mod
ul 8\no 4.py
Mangga
Duku
None

Sudah selesai.

--- Oleh L200210137 ---
>>>
```

Nomor 5 :

```
#NO.5
class Priorityqueue():
    def __init__(self):
        self.qlist=[]
    def isEmpty(self):
        return len(self)==0
    def __len__(self):
        return len(self.qlist)
    def enqueue(self,item,priority):
        entry = _PriorityQue(item,priority)
        self.qlist.append(entry)
    def dequeue(self):
        n = []
        for i in self.qlist:
            n.append(i.priority)
```

```
            print (self.qlist.pop(n.index(min(n))).item)
        def getFrontMost(self):
            return self.qlist[0]
        def getRearMost(self):
            return self.qlist[len(self.qlist)-1]

class _PriorityQue():
    def __init__(self,data,priority):
        self.item = data
        self.priority= priority

q = Priorityqueue()
q.enqueue("Triadmaja", 7)
q.enqueue("Bima", 1)
q.enqueue("Prak ASD Kelas E", 78)
q.enqueue("NIM L200210137", 9)
print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```

```
= RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Mod
ul 8\no5.py

Sudah selesai.

--- Oleh L200210137 ---
>>> q.dequeue()
Bima
>>> q.dequeue()
Triadmaja
>>> q.dequeue()
NIM L200210137
>>> q.dequeue()
Prak ASD Kelas E
>>> q.dequeue()
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    q.dequeue()
  File "C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul
8\no5.py", line 16, in dequeue
    print (self.qlist.pop(n.index(min(n))).item)
ValueError: min() arg is an empty sequence
>>>
```

Ln: 79 Col: 0