LAPORAN

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

(MODUL 8)

“ STACKS AND QUEUES ”

Disusun oleh :

NAMA : CINDI DILA APRILIANA

NIM : L200200106

KELAS : E

INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

TAHUN 2021/2022

1. Buatlah program untuk mengubah representasi suatu bilangan dari basis sepuluh ke basis dua.

- LATIHAN 1

Latihan1.py - C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma...

File Edit Format Run Options Window Help

```
#MODUL 8
#CINDI DILA APRILIANA_L200200106

#Latihan 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(), "Stack kosong. Tidak bisa diintip."
        return self.items[-1]
    def pop(self):
        assert not self.isEmpty(), "Stack kosong. Tidak bisa di-pop."
        return self.items.pop()
    def push(self,data):
        self.items.append(data)
```

- NOMOR 1

Nomor 1.py - C:\Users\MSI GF63\Documents\UMS\TUGAS SEMESTER 4\Praktikum Algoritma\MODUL 8\Nomc

File Edit Format Run Options Window Help

```
#MODUL 8
#CINDI DILA APRILIANA_L200200106

#Nomor 1

from Latihan1 import Stack

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

- Output

```
IDLE Shell 3.10.2
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 1
on win32
Type "help", "copyright", "credits" or "license()"
>>>
= RESTART: C:\Users\MSI GF63\Documents\UMS\TUGAS SEI
Nomor 1.py
>>> cetakHexa(12)
'C'
>>> cetakHexa(31)
'1F'
>>> cetakHexa(229)
'E5'
>>> cetakHexa(31519)
'7B1F'
>>>
```

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

```
Nomor 2.py - C:/Users/MSI GF63/Documents/UMS/TUGAS SEM
File  Edit  Format  Run  Options  Window  Help
#MODUL 8
#CINDI DILA APRILIANA_L200200106

#Nomor 2
from Latihan1 import *

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

- Output

```
IDLE Shell 3.10.2
File  Edit  Shell  Debug  Options  Windo
Python 3.10.2 (tags/v3.10.2:a
on win32
Type "help", "copyright", "c
>>>
= RESTART: C:/Users/MSI GF63/
Nomor 2.py
[0, 3, 6, 9, 12, 15]
>>>
```

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

- NOMOR 3

Nomor 3.py - C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Alg

File Edit Format Run Options Window Help

```
#MODUL 8
#CINDI DILA APRILIANA_L200200106

#Nomor 3

from Latihan1 import *

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

- Output

IDLE Shell 3.10.2

File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc,
on win32
Type "help", "copyright", "credits" c
>>>
= RESTART: C:/Users/MSI GF63/Document
Nomor 3.py
[0, 9, 12, 15]
>>>
```

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

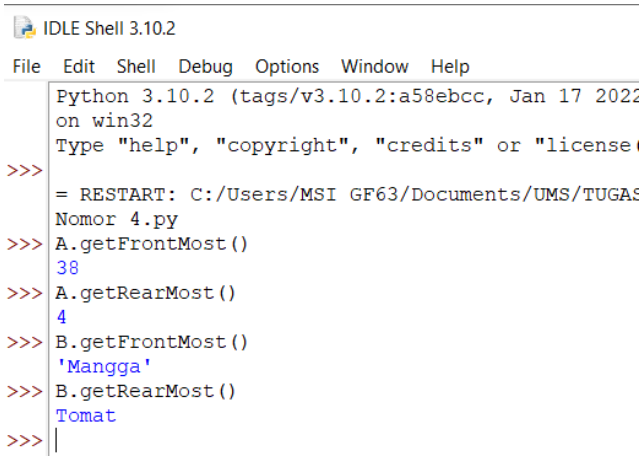
- NOMOR 4

Nomor 4.py - C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/

File Edit Format Run Options Window Help

```
#MODUL 8
#CINDI DILA APRILIANA_L200200106

#Nomor 4

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 sedang kosong"
        return self.qlist.pop(0)
    def getFrontMost(self):
        return self.qlist[0]
    def getRearMost(self):
        return self.qlist[-1]

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 getFrontMost(self):
        x = 0
        while self.qlist[x].priority != 0:
            x+=1
        return self.qlist[x].item
    def getRearMost(self):
        a = []
        for i in self.qlist:
            a.append(i.priority)
        print (self.qlist[a.index(max(a))].item)

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

A = Queue()
A.enqueue(38)
A.enqueue(12)
A.enqueue(45)
A.enqueue(23)
A.enqueue(4)

B = PriorityQueue()
B.enqueue("Jeruk", 3)
B.enqueue("Tomat", 5)
B.enqueue("Mangga", 0)
B.enqueue("Duku", 2)
B.enqueue("Pepaya", 1)
```

- Output



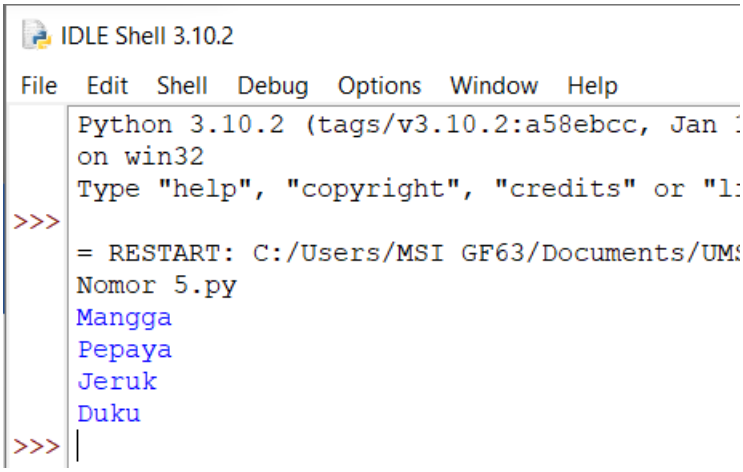
5. Pada class PriorityQueue di atas, metode dequeue() belum diimplementasikan. Tulis lah metode dequeue() ini dengan memperhatikan syarat-syarat seperti yang telah dicantumkan di halaman 81
    - NOMOR 5

Nomor 5.py - C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 8/Nomor...

File Edit Format Run Options Window Help

```
#MODUL 8
#CINDI DILA APRILIANA_L200200106

#Nomor 5

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 sedang kosong"
        a = []
        for i in self.qlist:
            a.append(i.priority)
        print (self.qlist.pop(a.index(min(a))).item)

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

S = PriorityQueue()
S.enqueue("Jeruk", 2)
S.enqueue("Tomat", 4)
S.enqueue("Mangga", 0)
S.enqueue("Duku", 3)
S.enqueue("Pepaya", 1)
S.dequeue()
S.dequeue()
S.dequeue()
S.dequeue()
```

- Output


IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan
on win32
Type "help", "copyright", "credits" or "l
>>>
= RESTART: C:/Users/MSI GF63/Documents/UM
Nomor 5.py
Mangga
Pepaya
Jeruk
Duku
>>>