PRAKTIKUM

ALGORITMA DAN STRUKTUR DATA

MODUL 8

Nama : NICKY JULYATRIKA SARI

NIM : L200200101

PROGRAM STUDI

INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

TAHUN 2021/2022

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

nomer 1.py - D:/MATKUL SMT 4/Praktikum ASD/modul 8/nomer 1.py (3.8.6)

File Edit Format Run Options Window Help

```
#Nicky Julyatrika Sari NIM L200200101
#Modul 8
#Nomer 1

class Stacks():
    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)
```

```
def cetakHexa(bil):
    x = Stacks()
    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
```

Python 3.8.6 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/MATKUL SMT 4/Praktikum ASD/modul 8/nomer 1.py ==========
>>> cetakHexa(12)
'C'
>>> cetakHexa(31)
'1F'
>>> cetakHexa(229)
'E5'
>>> cetakHexa(255)
'FF'
>>> cetakHexa(31519)
'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 ke lambang A, B, C, D, E, dan F

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

nomer 2.py - D:/MATKUL SMT 4/Praktikum ASD/modul 8/nomer 2.py (3.8.6)

File Edit Format Run Options Window Help

```
#Nicky Julyatrika Sari NIM L200200101
#Modul 8
#Nomer 2

class Stacks():
    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)


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

Python 3.8.6 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/MATKUL SMT 4/Praktikum ASD/modul 8/latihan 1.py =========
[0, 3, 6, 9, 12, 15]
>>>
```

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

nomer 3.py - D:/MATKUL SMT 4/Praktikum ASD/modul 8/nomer 3.py (3.8.6)

File Edit Format Run Options Window Help

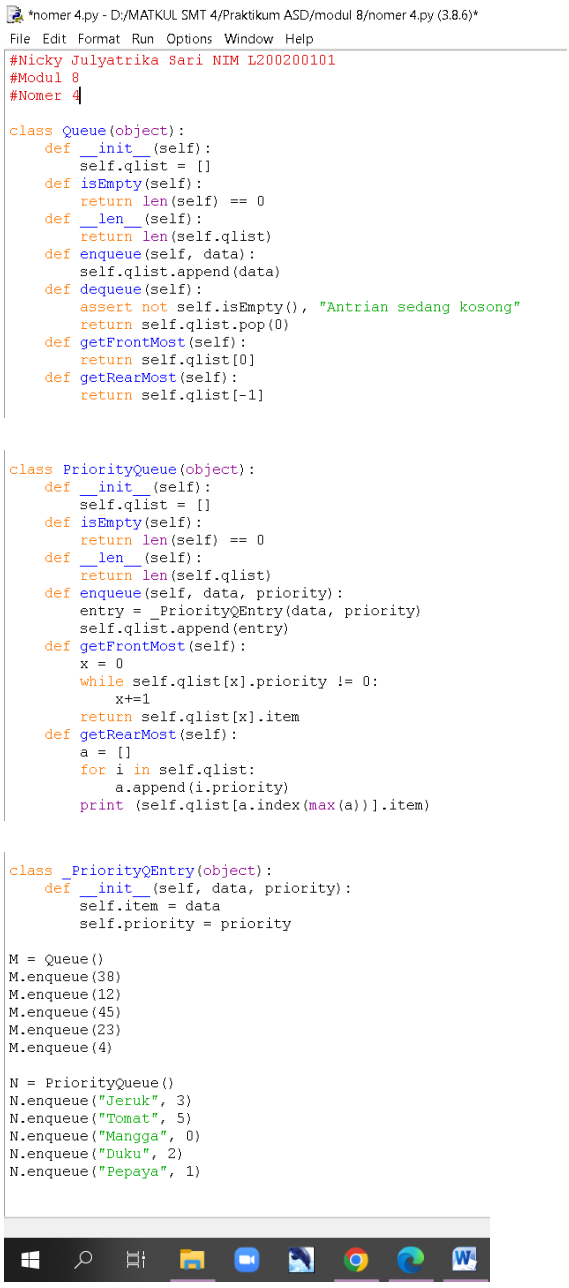
```
#Nicky Julyatrika Sari NIM L200200101
#Modul 8
#Nomer 3

class Stacks():
    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)

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

Python 3.8.6 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/MATKUL SMT 4/Praktikum ASD/modul 8/nomer 3.py ==========
[0, 9, 12, 15]
>>>
```

- Queues
  4. Tulis dua metode berikut ke class Queue dan class PriorityQueue di atas •
     a. Metode untuk mengetahui item yang paling depan tanpa menghapusnya def getFrontMost(self)
     b. Metode untuk mengetahui item yang paling belakang tanpa menghapusnya def getRearMost(self)



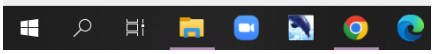
```
*nomer 4.py - D:/MATKUL SMT 4/Praktikum ASD/modul 8/nomer 4.py (3.8.6)*
File  Edit  Format  Run  Options  Window  Help
#Nicky Julyatrika Sari NIM L200200101
#Modul 8
#Nomer 4

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

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

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

  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

nomer 5.py - D:/MATKUL SMT 4/Praktikum ASD/modul 8/nomer 5.py (3.8.6)

File Edit Format Run Options Window Help

```
#Nicky Julyatrika Sari NIM L200200101
#Modul 8
#Nomer 2

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()
```



Python 3.8.6 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/MATKUL SMT 4/Praktikum ASD/modul 8/nomer 5.py ==========
Mangga
Pepaya
Jeruk
Duku
>>>
```