

Minggu, 4 Agustus 2019

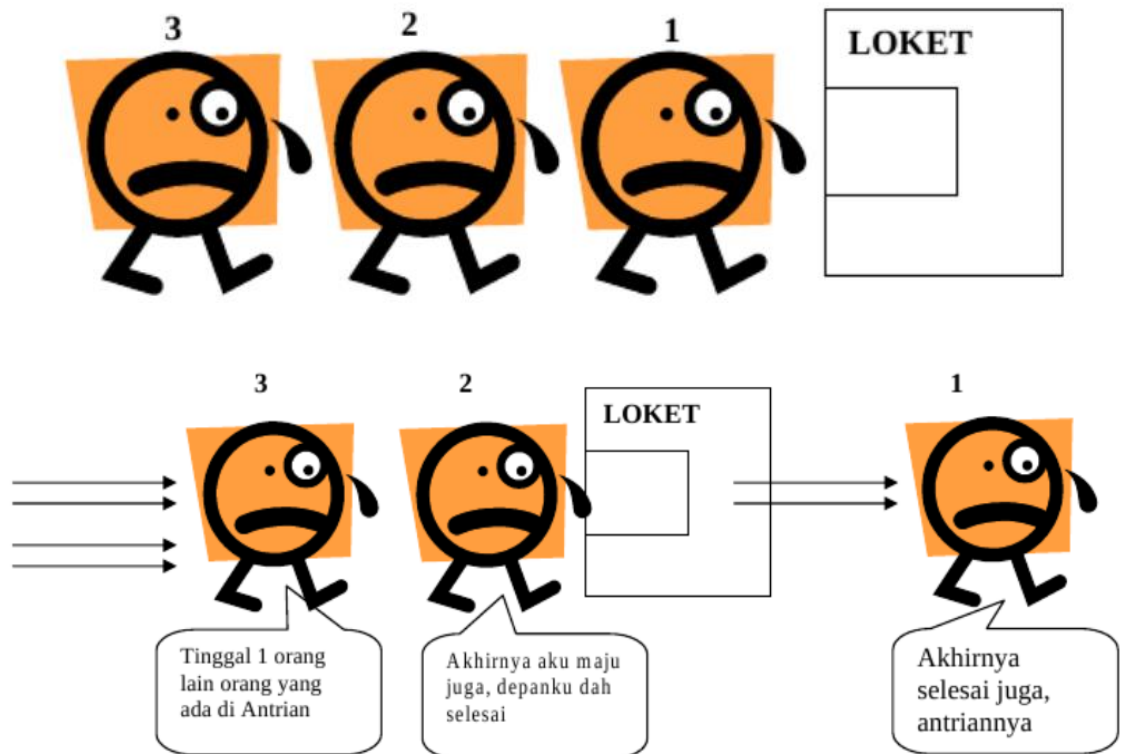
LAPORAN KEGIATAN (QUEUE)

A. DASAR TEORI

1. *Queue* (Antrian) adalah suatu kumpulan data yang mana penambahan data / elemen hanya dapat dilakukan pada sisi belakang sedangkan penghapusan / pengeluaran elemen dilakukan pada sisi depan. Berbeda dengan *stack*, prinsip yang digunakan dalam antrian adalah FIFO (First In First Out). Dengan kata lain, urutan keluar elemen akan sama dengan urutan masuknya.

- Enqueue adalah memasukkan satu elemen dari suatu antrian.
- Dequeue adalah mengeluarkan satu elemen dari suatu antrian.

Ilustrasi Queue:



B. PEMBAHASAN

1. Pada praktikum [pertama](#) ini akan mencoba membuat sebuah program dimana program ini telah jadi namun *source code* harus diubah menjadi ke dalam bahasa pemrograman Python 3.

```
1  class Queue:
2      que_array = []
3      head = 0
4      tail = 0
5      n_items = 0
6      max_size = 0
7
8      def __init__(self, x):
9          self.max_size = x
10         self.que_array = [0] * self.max_size
11         self.head = 0
12         self.tail = -1
13         self.n_items = 0
14
15     def enqueue(self, n):
16         self.tail += 1
17         self.que_array[self.tail] = n
18         self.n_items += 1
```

```

19
20     def dequeue(self):
21         tmp = self.que_array[self.head]
22         self.head += 1
23         self.n_items -= 1
24         return tmp
25
26     def peek_head(self):
27         if not self.is_empty() > 0:
28             return self.que_array[self.head]
29         else:
30             return None
31
32     def is_empty(self):
33         return self.n_items == 0
34
35     def is_full(self):
36         return self.tail == self.max_size - 1

```

```

37
38     def size(self):
39         return self.n_items
40
41 class RezkyYustisioHadiPratam_Queue_10:
42     the_queue = Queue(5)
43     the_queue.enqueue(10)
44     the_queue.enqueue(20)
45     the_queue.enqueue(30)
46     the_queue.enqueue(40)
47
48     the_queue.dequeue()
49     the_queue.dequeue()
50     the_queue.dequeue()
51
52     the_queue.enqueue(50)
53     the_queue.enqueue(60)
54     the_queue.enqueue(70)
55     the_queue.enqueue(80)
56     the_queue.enqueue(90)

```

```

57
58     while( not the_queue.is_empty()):
59         n = the_queue.dequeue()
60         print(n, end=" ")
61     print()

```

Source Code Program

<Penjelasan dari source code di atas>

- Pada baris nomor 1, berfungsi untuk membuat sebuah class dengan nama **Queue**.
- Pada baris nomor 2, berfungsi untuk menampung nilai elemen-elemen.
- Pada baris nomor 3, berfungsi untuk menampung nilai elemen paling pertama.
- Pada baris nomor 4, berfungsi untuk menampung nilai elemen paling akhir.
- Pada baris nomor 5, berfungsi untuk menampung nilai jumlah elemen pada antrian.
- Pada baris nomor 6, berfungsi untuk menampung nilai maksimum untuk antrian.
- Pada baris nomor 8 sampai 13, berfungsi untuk insialisasi sekaligus menerima parameter **x** sebagai jumlah elemen yang bisa ditampung untuk antrian. Selanjutnya mengeset nilai **max_size**, **que_array**, **head**, **tail**, **n_items**.
- Pada baris nomor 15 sampai 18 berfungsi untuk memasukan satu elemen ke dalam antrian.
- Pada baris nomor 20 sampai 24, berfungsi untuk mengeluarkan satu elemen dari antrian.

- Pada baris nomor 26 sampai 30, berfungsi untuk mengetahui elemen yang paling pertama masuk pada antrian.
- Pada baris nomor 32 sampai 33, berfungsi untuk membuktikan apakah antrian tersebut kosong atau tidak.
- Pada baris nomor 35 sampai 36, berfungsi untuk membuktikan apakah antrian tersebut sudah penuh atau belum.
- Pada baris nomor 38 sampai 39, berfungsi untuk mengetahui jumlah dari elemen pada antrian.
- Pada baris nomor 41, berfungsi untuk membuat sebuah class dengan nama **RezkyYustisioHadiPratama_Queue_10**.
- Pada baris nomor 42, berfungsi untuk membuat objek dari class Queue.
- Pada baris nomor 43 sampai 46, berfungsi untuk memasukan elemen ke dalam antrian.
- Pada baris nomor 48 sampai 50, berfungsi untuk mengeluarkan elemen dari antrian.
- Pada baris nomor 52 sampai 56, berfungsi untuk memasukan elemen ke dalam antrian.

2. Pada praktikum [kedua](#) ini akan menjalankan program pada praktikum pertama, kemudian menjelaskan hasil program tersebut.

```

IndexError: list assignment index out of range
>>> |

```

Output Program

* Penjelasan dari output program di atas

- a. Pada output di atas muncul pesan "IndexError: list assignment index out of range" yang artinya elemen yang di masukan ke dalam array tersebut melebihi batas jumlah tampung array. Faktor yang menyebabkan index

out of range adalah ketika meng-enqueue melebihi batas jumlah tampung array.

3. Pada praktikum [ketiga](#) ini akan mengembangkan program pada praktikum sebelumnya dimana ada penambahan *function* yang berfungsi untuk menghapus satu elemen di dalam antrian.

```
41     def delete_element(self, j):
42         self.que_array.remove(j)
43         return j
```

Source Code Program

<Penjelasan dari source code di atas>

- Pada baris nomor 41, berfungsi membuat sebuah *function* dengan nama **delete_element** yang menerima parameter **j**. Parameter **j** adalah angka/element yang akan dihapus di dalam antrian.
 - Pada baris nomor 42, berfungsi untuk melakukan penghapusan terhadap elemen di dalam antrian dengan menggunakan fungsi **remove**.
4. Pada praktikum [keempat](#) ini akan mengembangkan program pada praktikum sebelumnya dimana ada penambahan *function* yang berfungsi untuk mencari suatu elemen di dalam antrian.

```
45     def search_element(self, j):
46         if j in self.que_array:
47             print(str(j)+ " Ditemukan")
48         else:
49             print(str(j)+" tidak ditemukan")
```

Source Code Program

<Penjelasan dari source code di atas>

- Pada baris nomor 45, berfungsi membuat sebuah *function* dengan nama **search_element** yang menerima parameter **j**. Parameter **j** adalah angka/elemen yang ingin dicari di dalam antrian.
- Pada baris nomor 46 sampai 49, berfungsi untuk membuat suatu perbandingan dimana jika angka terdapat di dalam antrian maka akan menampilkan pesan angka ditemukan. Jika tidak maka akan menampilkan pesan angka tidak ditemukan.

C. KESIMPULAN

Queue atau antrian adalah struktur data yang memiliki sifat FIFO (Last In First Out) yang artinya semua objek ketika pertama kali masuk akan keluar paling awal. Di ibaratkan seperti antrian di SBPU. Pengendara yang datang dengan antrian paling pertama akan dilayani terlebih dahulu.

D. DAFTAR PUSTAKA

Wibowo,Fitri. __, *Queue*, [pdf]