

PERTEMUAN 13

QUEUE

A. TUJUAN PEMBELAJARAN

Setelah mempelajari materi pertemuan diharapkan mahasiswa mengerti dan paham queue, proses FIFO, membuat dan menghapus item data antrian dan mampu membuat program queue dalam Python

B. URAIAN MATERI

1. Queue

Kata antrian dalam bahasa Inggris untuk baris untuk "mengantri" berarti mengantre. Dalam ilmu komputer, antrian adalah struktur data yang agak berbeda seperti tumpukan, kecuali bahwa dalam antrian, item pertama yang dimasukkan adalah yang pertama jadi dihapus (First-In-First-Out, FIFO), sementara dalam tumpukan, seperti yang telah terlihat, item terakhir dimasukkan adalah yang pertama akan dihapus (LIFO). Antrian berfungsi seperti baris di film: Orang pertama yang bergabung di belakang garis adalah orang pertama yang mencapai garis depan antrian dan beli tiket. Orang terakhir yang mengantre adalah orang terakhir yang membeli tiket (atau — jika pertunjukan terjual habis — gagal membeli tiket).



Gambar 13. 1 Antrian Orang

Antrian digunakan sebagai alat pemrogram sebagai tumpukan. Contoh di mana sebuah antrian membantu mencari grafik, digunakan untuk membuat model situasi dunia nyata seperti orang yang mengantri di bank, pesawat menunggu lepas landas, atau data paket menunggu untuk dikirim melalui Internet.

Ada berbagai antrian yang melakukan tugasnya di komputer (atau jaringan). Ada antrian printer tempat pekerjaan cetak menunggu printer akan tersedia. Antrian juga menyimpan data keystroke saat mengetik di papan ketik. Dengan cara ini, jika menggunakan pengolah kata tetapi komputer digunakan sebentar melakukan hal lain saat menekan tombol, tombol tidak akan hilang; itu menunggu di antri hingga pengolah kata memiliki waktu untuk membacanya. Menggunakan antrian menjamin penekanan tombol tetap berurutan sampai dapat diproses.

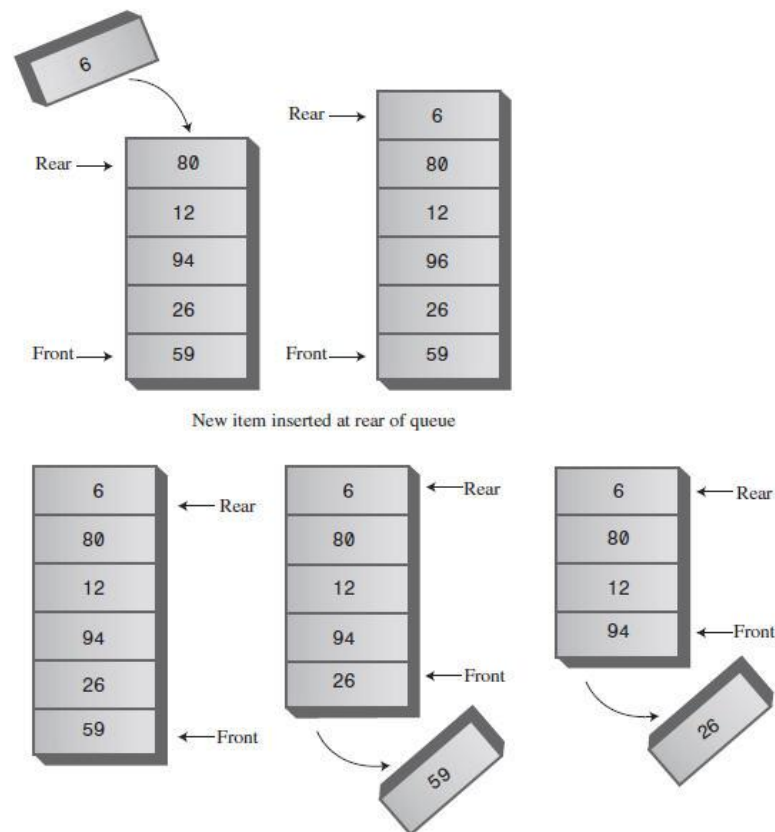
Applet ini mendemonstrasikan antrian berdasarkan array. Ini adalah pendekatan umum, meskipun daftar tertaut juga biasa digunakan untuk mengimplementasikan antrian. Dua operasi antrian dasar adalah memasukkan item, yang ditempatkan di belakang antrian, dan menghapus item, yang diambil dari depan antrian. Ini adalah mirip dengan seseorang yang bergabung dengan barisan belakang penonton film dan, setelah tiba di barisan depan dan membeli tiket, melepaskan dirinya dari depan garis.

Istilah penyisipan dan penghapusan dalam tumpukan cukup standar; semua orang bilang dorong dan muncul. Standardisasi belum berkembang sejauh ini dengan antrian. Sisipkan juga disebut put atau add atau enqueue, sedangkan remove bisa disebut delete atau get atau deque. Bagian belakang queue, tempat item dimasukkan, juga disebut back atau tail atau end. Bagian depan, dimana barang dilepas, bisa juga disebut kepala. Kami akan menggunakan istilah sisipan, lepas, depan, dan belakang.

2. Insert Button Antrian

Dengan menekan berulang kali tombol Ins di applet antrian, dapat menyisipkan item baru. Setelah penekanan pertama, akan diminta memasukkan nilai kunci untuk item baru ke dalam bidang teks Angka; ini harus berupa angka dari 0 hingga 999. Selanjutnya penekanan akan memasukkan item dengan kunci ini di belakang antrian dan menaikkan panah belakang sehingga menunjuk ke item baru.

Tombol Hapus Demikian pula, dapat menghapus item di depan antrian menggunakan tombol Rem. Item dihapus, nilai item disimpan di bidang Angka (sesuai dengan metode remove () mengembalikan nilai), dan panah depan bertambah. Dalam applet, sel yang menahan item yang dihapus berwarna abu-abu untuk menunjukkan bahwa item tersebut hilang. Secara normal implementasi, itu akan tetap dalam memori tetapi tidak akan dapat diakses karena Front telah melewatinya. Operasi masukkan dan hapus ditunjukkan pada Gambar 12.2.

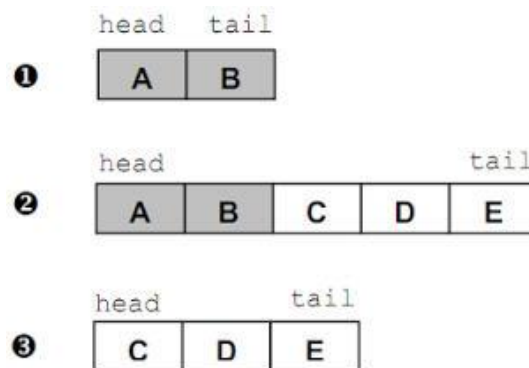


Gambar 13. 2 Operasi Metode Kelas Antrian

Dalam struktur data, antrian sedikit berbeda dengan stack. Jika dalam stack, data yang pertama masuk adalah yang paling akhir keluar, namun dalam queue data yang paling awal masuk adalah yang paling awal keluar. Konsep dari Queue adalah LIFO (Last In First Out).

Contoh dalam kehidupan, isalnya antrian dibioskop. pada saat mengantri, orang yang pertama kali mengantri akan mendapat pelayanan pertama dan yang pertama yang akan keluar dari antrian

Penggambaran dari Queue dalam struktur data adalah sebagai berikut :



Gambar 13. 3 Ilustrasi Queue dalam Struktur Data

Pada saat menempatkan elemen pada ujung (tail) dari queue disebut dengan *enqueue*. Pada saat memindahkan elemen dari kepala (head) ke dalam queue disebut dengan *dequeue*. Karakteristik penting dari queue adalah:

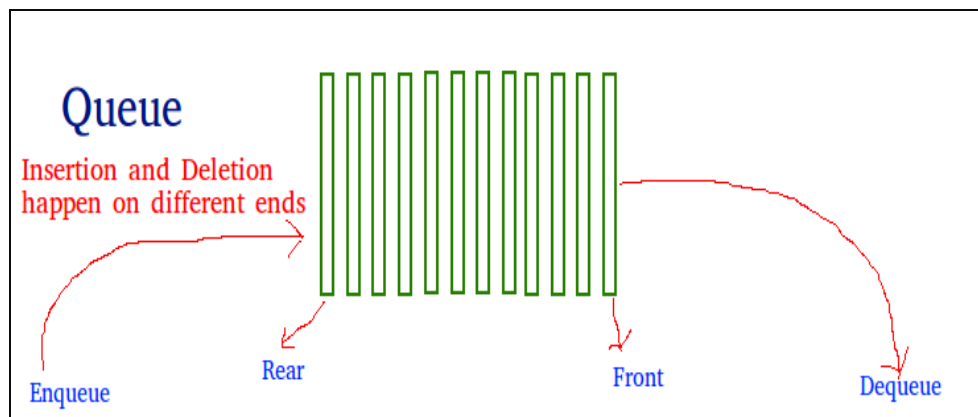
- Element antrian yaitu item-item data yang terdapat di elemen antrian
- Front (element terdepan di antrian)
- Rear (element terakhir di antrian)
- Jumlah elemen pada antrian (Count)
- Status antrian

Kondisi antrian yang menjadi perhatian adalah :

- Penuh :** Bila elemen pada antrian mencapai kapasitas maksimum antrian. Pada kondisi ini, tidak mungkin dilakukan penambahan ke antrian. Penambahan elemen menyebabkan kondisi kesalahan Overflow.
- Kosong :** Bila elemen pada antrian mencapai kapasitas maksimum antrian. Pada kondisi ini, tidak mungkin dilakukan penambahan ke antrian. Penambahan elemen menyebabkan kondisi kesalahan Overflow.

3. Menambahkan Elemen

Dalam contoh ini membuat kelas antrian di mana kita menerapkan metode First-in-First-Out. Penggunaan metode sisipkan bawaan untuk menambahkan elemen data. seperti tumpukan, antrian adalah struktur data linier yang menyimpan item dengan cara First In First Out (FIFO). Dengan antrian, item yang paling sedikit ditambahkan akan dihapus terlebih dahulu. Contoh antrian yang baik adalah antrian konsumen di mana konsumen yang pertama dilayani terlebih dahulu.



Gambar 13. 4 FIFO dalam Antrian

Operasi yang terkait dengan antrian adalah:

- Enqueue: Menambahkan item ke antrian. Jika antrian penuh, maka dikatakan sebagai kondisi Overflow – Kompleksitas Waktu: $O(1)$
- Dequeue: Menghapus item dari antrian. Item yang muncul dalam urutan yang sama di mana mereka didorong. Jika antrian kosong, maka dikatakan sebagai kondisi Underflow – Kompleksitas Waktu: $O(1)$
- Depan: Dapatkan item depan dari antrian – Kompleksitas Waktu: $O(1)$
- Belakang: Dapatkan item terakhir dari antrian – Kompleksitas Waktu: $O(1)$

4. Implementasi Menggunakan Antrian.

Queue adalah modul built-in Python yang digunakan untuk mengimplementasikan antrian. Antrian (maxsize) menginisialisasi variabel ke ukuran maksimal maxsize. Maxsize dari nol '0' berarti antrian yang tak terbatas. Antrian ini mengikuti aturan FIFO. Ada berbagai fungsi yang tersedia dalam modul ini:

- a. `maxsize` – Jumlah item yang diizinkan dalam antrian.
- b. `kosong()` – Kembalikan `True` jika antrian kosong, Salah sebaliknya.
- c. `full()` – Return `True` jika ada item maksimal dalam antrian. Jika antrian disialalisasi dengan `maxsize = 0` (default), maka penuh () tidak pernah mengembalikan `True`.
- d. `get()` – Hapus dan kembalikan item dari antrian. Jika antrian kosong, tunggu sampai item tersedia.
- e. `get_nowait()` – Kembalikan item jika ada yang segera tersedia, yang lain meningkatkan `QueueEmpty`.
- f. `put (item)` – Masukkan item ke dalam antrian. Jika antrian penuh, tunggu sampai slot gratis tersedia sebelum menambahkan item.
- g. `put_nowait (item)` – Masukkan item ke dalam antrian tanpa pemblokiran. Jika tidak ada slot gratis yang segera tersedia, naikan `QueueFull`.
- h. `qsize()` – Kembalikan jumlah item dalam antrian.

```

1 # demonstrate implementation of
2 # queue using queue module
3
4 from queue import Queue
5 # Initializing a queue
6 q = Queue(maxsize = 3)
7
8 # qsize() give the maxsize
9 # of the Queue
10 print(q.qsize())
11
12 # Adding of element to queue
13 q.put('a')
14 q.put('b')
15 q.put('c')
16
17 # Return Boolean for Full
18 # Queue
19 print("\nFull: ", q.full())
20
21 # Removing element from queue
22 print("\nElements dequeued from the queue")
23 print(q.get())
24 print(q.get())
25 print(q.get())
26
27 # Return Boolean for Empty
28 # Queue
29 print("\nEmpty: ", q.empty())
30
31 q.put(1)
32 print("\nEmpty: ", q.empty())
33 print("Full: ", q.full())
34
35 # This would result into Infinite
36 # Loop as the Queue is empty.
37 # print(q.get())
38

```

Hasil:

0

Full: True

Elements dequeued from the queue

a

b

c

Empty: True

Empty: False

Full: False

5. Cara Menggunakan Antrian di Python

Untuk mulai membangun antrian Python, diperlukan mengimpor modul Python terlebih dahulu:queue

Import queue

Semua versi yang lebih baru memiliki modul ini tersedia untuk digunakan. Hal ini memungkinkan untuk menerapkan antrian multithreadingPython:

- a. Untuk menambahkan elemen ke antrian, gunakan . Ini disebut operasi enqueue.put()
- b. Untuk menghapus elemen dari antrian, gunakan . Ini disebut operasi dequeue.get()
- c. Prinsip FIFO (First In, First Out) berarti elemen pertama yang dimasukkan juga akan menjadi yang pertama dihapus.

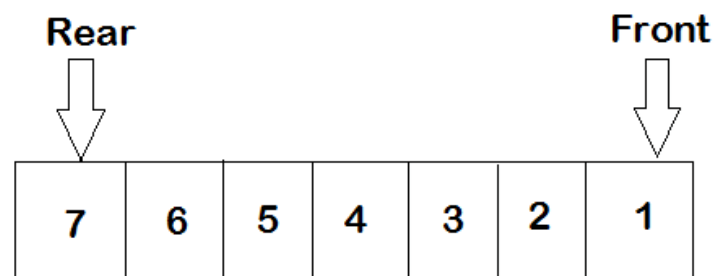
Dengan menggunakan argumen, dapat memilih untuk membatasi jumlah elemen dalam antrian. Untuk membiarkannya tidak terbatas, definisikan sebagai atau angka negatif.maxsize0

- a. Untuk menyederhanakan bekerja dengan antrian prioritas, ikuti pola dan gunakan untuk menentukan prioritas.number, elementnumber
- b. Untuk membuat antrian multiprocessing Python (sebagai lawan multithreading), gunakan fungsi untuk modul.multiprocessing.Queue()multiprocessin

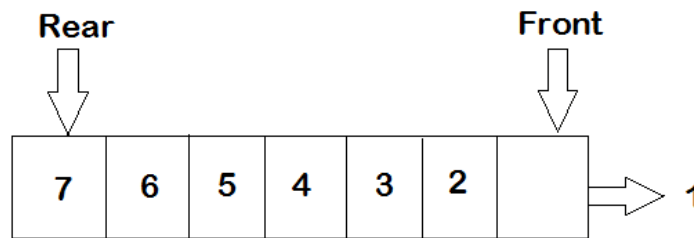
Antrian adalah wadah yang menyimpan data. Data yang dimasukkan pertama akan dihapus terlebih dahulu, dan karenanya antrian juga disebut "First in First Out" (FIFO). Antrian memiliki dua ujung depan dan belakang. Item dimasukkan dari belakang dan dikeluarkan dari sisi depan.

6. Cara Kerja Queue

Antrian dapat dengan mudah dibandingkan dengan contoh dunia nyata garis orang yang menunggu dalam antrian di loket tiket, orang yang berdiri lebih dulu akan mendapatkan tiket terlebih dahulu, diikuti oleh orang berikutnya dan sebagainya. Logika yang sama berlaku untuk struktur data antrian juga. Berikut adalah representasi diagram dari antrian:



- a. Bagian Belakang mewakili titik di mana item dimasukkan ke dalam antrian. Dalam contoh ini, 7 adalah nilai untuk itu.
- b. Front mewakili titik di mana item dari antrian akan dihapus. Jika menghapus item dari antrian, elemen pertama yang akan didapatkan adalah 1, seperti yang ditunjukkan pada gambar.
- c. Item 1 adalah yang pertama dimasukkan dalam antrian, dan saat menghapusnya adalah yang pertama keluar. Oleh karena itu antrian disebut FIRST IN FIRST OUT (FIFO)



Dalam antrian, item dihapus secara berurutan dan tidak dapat dihapus dari antaranya, tidak dapat menghapus item 5 secara acak dari antrian, untuk melakukan itu dimana harus menghapus semua item sebelum 5. Item dalam antrian akan dihapus dalam urutan yang dimasukkan. Jenis Antrian di Python ada dua jenis antrian di Python:

- a. Pertama di First out Queue: Untuk ini, elemen yang keluar pertama akan menjadi yang pertama keluar. Untuk bekerja dengan FIFO, harus menghubungi kelas Antrian() dari modul antrian.
- b. Terakhir di First out Queue: Di sini, elemen yang dimasukkan terakhir akan menjadi yang pertama keluar. Untuk bekerja dengan LIFO, dimana harus menghubungi kelas LifoQueue() dari modul antrian.

7. Menambahkan Item Antrian

Mari kerjakan contoh untuk menambahkan item dalam antrian. Untuk mulai bekerja dengan antrian, pertama-tama impor antrian modul, seperti yang ditunjukkan pada contoh di bawah ini. Untuk menambahkan item, dapat menggunakan metode put() seperti yang ditunjukkan pada contoh:

```
import queue
```

```
q1 = queue.Queue()
```

```
q1.put(10) #this will add item 10 to the queue.
```

Secara default, ukuran antrian tidak terbatas dan dapat menambahkan sejumlah item ke dalamnya. Jika ingin menentukan ukuran antrian yang sama dapat dilakukan sebagai berikut:

```
import queue
```

```
q1 = queue.Queue(5) #The max size is 5.
```

```
q1.put(1)
q1.put(2)
q1.put(3)
q1.put(4)
q1.put(5)
print(q1.full()) # will return true.
```

Hasil:

True

Sekarang ukuran antrian adalah 5, dan tidak akan mengambil lebih dari 5 item, dan metode `q1.full()` akan kembali benar. Menambahkan item lagi tidak akan mengeksekusi kode lebih jauh.

8. Menghapus Item Antrian

Untuk menghapus item dari antrian, dimana dapat menggunakan metode yang disebut `get()`. Metode ini memungkinkan item dari antrian saat dipanggil. Contoh berikut menunjukkan cara menghapus item dari antrian.

```
import queue
q1 = queue.Queue()
q1.put(10)
item1 = q1.get()
print('The item removed from the queue is ', item1)
```

Hasil:

The item removed from the queue is 10

Ringkasan dalam proses diatas adalah:

- Antrian adalah wadah yang menyimpan data. Ada dua jenis Queue, FIFO, dan LIFO.
- Untuk FIFO (First in First out Queue), elemen yang masuk pertama akan menjadi yang pertama keluar.

- c. Untuk LIFO (Last in First out Queue), elemen yang dimasukkan terakhir akan menjadi yang pertama keluar.
- d. Item dalam antrian ditambahkan menggunakan metode put (item).
- e. Untuk menghapus item, metode get() digunakan

9. Penerapkan Antrian di Python

Sangat mudah untuk bekerja dengan antrian di python. Berikut adalah langkah langkah yang harus diikuti untuk memanfaatkan antrian dalam kode :

- a. Langkah 1) Anda hanya perlu mengimpor modul antrian, seperti yang ditunjukkan di bawah ini:

```
import queue
```

Modul ini tersedia secara default dengan python, dan Anda tidak memerlukan instalasi tambahan untuk mulai bekerja dengan antrian. Ada 2 jenis antrian FIFO (pertama di first out) dan LIFO (terakhir di first out).

- b. Langkah 2) Untuk bekerja dengan antrian FIFO, hubungi kelas Antrian menggunakan modul antrian yang diimpor seperti yang ditunjukkan di bawah ini:

```
import queue
```

```
q1 = queue.Queue()
```

- c. Langkah 3) Untuk bekerja dengan antrian LIFO, hubungi kelas LifoQueue() seperti yang ditunjukkan di bawah ini:

```
import queue
```

```
q1 = queue.LifoQueue()
```

Metode yang tersedia di dalam kelas Queue dan LifoQueue

Berikut ini adalah metode penting yang tersedia di dalam kelas Queue dan LifoQueue:

- a. put(item): Ini akan menempatkan item di dalam antrian.
- b. dapatkan(): Ini akan mengembalikan item dari antrian.
- c. kosong(): Ini akan kembali benar jika antrian kosong dan salah jika item hadir.

- d. `qsize()`: mengembalikan ukuran antrian.
- e. `penuh()`: kembali benar jika antrian penuh, jika tidak salah.

Dalam kasus pertama di pertama keluar, elemen yang pergi pertama akan menjadi yang pertama untuk keluar. Ada berbagai cara untuk menerapkan antrian di Python. Artikel ini mencakup penerapan antrian menggunakan struktur data dan modul dari perpustakaan Python. Antrian di Python dapat diimplementasikan dengan cara berikut:

- a. daftar
- b. `collections.deque`
- c. `queue`

10. Implementasi Menggunakan List

List adalah struktur data built-in Python yang dapat digunakan sebagai antrian. Alih-alih `enqueue()` dan `dequeue()`, fungsi `append()` dan `pop()` digunakan. Namun, daftar cukup lambat untuk tujuan ini karena memasukkan atau menghapus elemen di awal membutuhkan pergeseran semua elemen lain dengan satu, membutuhkan waktu $O(n)$.

```
1 # Python program to
2 # demonstrate queue implementation
3 # using list
4 # Initializing a queue
5 queue = []
6 # Adding elements to the queue
7 queue.append('a')
8 queue.append('b')
9 queue.append('c')
10
11 print("Initial queue")
12 print(queue)
13
14 # Removing elements from the queue
15 print("\nElements dequeued from queue")
16 print(queue.pop(0))
17 print(queue.pop(0))
18 print(queue.pop(0))
19
20 print("\nQueue after removing elements")
21 print(queue)
22
```

Hasil:

Initial queue

['a', 'b', 'c']

Elements dequeued from queue

a

b

c

Queue after removing elements

[]

11. Implementasi Menggunakan Collections.Deque

Antrian di Python dapat diimplementasikan menggunakan kelas deque dari modul koleksi. Deque lebih disukai daripada daftar dalam kasus-kasus di mana kita membutuhkan tambahan dan operasi pop yang lebih cepat dari kedua ujung wadah, karena deque memberikan kompleksitas waktu $O(1)$ untuk operasi tambahan dan pop dibandingkan dengan daftar yang menyediakan $O(n)$ kompleksitas waktu. Alih-alih enqueue dan deque, fungsi `append()` dan `popleft()` digunakan.

```
1 # using collections.dequeue
2 from collections import deque
3
4 # Initializing a queue
5 q = deque()
6
7 # Adding elements to a queue
8 q.append('a')
9 q.append('b')
10 q.append('c')
11
12 print("Initial queue")
13 print(q)
14
15 # Removing elements from a queue
16 print("\nElements dequeued from the queue")
17 print(q.popleft())
18 print(q.popleft())
19 print(q.popleft())
20
21 print("\nQueue after removing elements")
22 print(q)
23
24 # Uncommenting q.popleft()
25 # will raise an IndexError
26 # as queue is now empty
```

Hasil:

Initial queue

deque(['a', 'b', 'c'])

Elements dequeued from the queue

a

b

c

Queue after removing elements

deque([])

Contoh Program Queue di python :

```

1 import os
2 import Queue
3
4 class myQueue:
5     def __init__(self):
6         self.items = Queue.Queue()
7
8     # Memeriksa apakah queue dalam keadaan kosong
9     def isEmpty(self):
10         return self.items.empty()
11     # Menambah data ke queue
12     def qPut(self, item):
13         self.items.put(item)
14     # Mengeluarkan data dari queue
15     def qGet(self):
16         if not self.items.empty():
17             return self.items.get()
18         else:
19             return "empty"
20     # Menghitung panjang queue
21     def size(self):
22         return self.items.qsize()
23
24     # Main menu aplikasi
25     def mainmenu(self):
26         pilih = "y"
27         while (pilih == "y"):
28             os.system("clear")
29             print("=====")
30             print("| Menu aplikasi queue |")
31             print("=====")
32             print("1. Put objek")
33             print("2. Get objek")
34             print("3. Cek Empty")
35             print("4. Panjang dari queue")
36             print("=====")
37             pilihan=str(input(("Silakan masukan pilihan anda: ")))
38             if(pilihan=="1"):

```

```

37     pilihan=str(input(("Silakan masukan pilihan anda: ")))
38     if(pilihan=="1"):
39         os.system("clear")
40         obj = str(input("Masukan objek yang ingin anda tambahkan: "))
41         self.qPut(obj)
42         print("Object "+obj+" telah ditambahkan")
43         x = raw_input("")
44     elif(pilihan=="2"):
45         os.system("clear")
46         temp = self.qGet()
47         if temp != "empty":
48             print("Objek "+temp+" dihapus")
49         else:
50             print("Queue kosong")
51         x = raw_input("")
52     elif(pilihan=="3"):
53         os.system("clear")
54         print(self.isEmpty())
55         x = raw_input("")
56     elif(pilihan=="4"):
57         os.system("clear")
58         print("Panjang dari queue adalah: "+str(self.size()))
59         x = raw_input("")
60     else:
61         pilih="n"
62
63 if __name__ == "__main__":
64     q=myQueue()
65     q.mainmenu()
66

```

Hasil jika di run adalah :

=====

| Menu aplikasi queue |

=====

1. Put objek
2. Get objek
3. Cek Empty
4. Panjang dari queue

=====

Antrian dalam kehidupan kita sehari-hari saat kita menunggu layanan. Struktur data antrian aslo berarti sama di mana elemen data diatur dalam antrian. Keunikan antrian terletak pada cara item ditambahkan dan dihapus. Item diperbolehkan di ujung tetapi dihapus dari ujung yang lain. Jadi ini adalah metode first-in-first out. Antrian dapat diimplementasikan menggunakan daftar python di mana dapat menggunakan metode sisipkan () dan pop() untuk menambahkan dan menghapus elemen. Tidak ada penyisipan elemen data selalu ditambahkan di akhir antrian.

C. LATIHAN SOAL

1. Jelaskan yang anda ketahui mengenai antrian!
2. Jelaskan operasi push dalam sebuah antrian!
3. Jelaskan operasi POP dalam sebuah antrian!
4. Jelaskan Bagaimana pengecekan kondisi diawal antrian!
5. Buatlah contoh implementasi antrian dengan bahasa python!

D. REFERENSI

- Amanulhaq, A. A. (2021). Implementasi Algoritma Image Hashing Dan Hamming Distance Untuk Deteksi Kemiripan Gambar. *Repository Its*.
- Basant Agarwal, B. B. (2018). Hand-On Data Structures And Algorithms With Python. London: Packt Publishing.
- Emi Sita Eriana, A. Z. (2021). Penerapan Metode Personal Extreme Programming Dalam Perancangan Aplikasi Pemilihan Ketua Hmsi Dengan Weighted Product. *Jurnal Ilmu Komputer*, 27-32.
- Emi Sita Eriana, A. Z. (2021). Praktikum Algoritma Dan Pemrograman. Tangerang Selatan: Unpam Press.
- Jodi, U. R. (2020). Algoritma Dan Struktur Data.
- Mohamad Aslam Katahman, M. F. (2021). Pembangunan Aplikasi Realiti Terimbu Untuk Pengenalan Struktur Data. *Information Technology And Computer Science*.

- Nasrullah, A. H. (2021). Implementasi Algoritma Decision Tree Untuk Klasifikasi Produk Laris. *Jurnal Ilmiah Ilmu Komputer I*.
- Peng Qi, Y. Z. (2020). Stanza: A Python Natural Language Processing Toolkit For Many Human Languages.
- Pradana Setialana, T. B. (2017). Pencarian Hubungan Kekerabatan Pada Struktur Data Genealogy Dalam Graph Databas.
- Ranny Meilisa, D. P. (2020). Model Pembelajaran Flipped Classroom Pada Mata Kuliah Algoritma Dan Struktur Data. *Jurnal Ilmiah Pendidikan Dan Pembelajaran (Jipp)*.
- Revanza, M. G. (2020). Struktur Data Dan Bahasa Pemrograman.