MODUL PERKULIAHAN

TFC251 – Praktikum Struktur Data

Queue Dalam Python

Penyusun Modul	:	Suamanda Ika Novichasari, M.Kom
Minggu/Pertemuan	:	4
Sub-CPMK/Tujuan Pembelajaran	:	1. Mahasiswa mampu menerapkan konsep queue pada bahasa pemrograman python
Pokok Bahasan	:	1. Queue

Program Studi Teknologi Informasi (S1)
Fakultas Teknik
Universitas Tidar
Tahun 2024

Materi 4
QUEUE DALAM PYTHON

Petunjuk Praktikum :

- Cobalah semua contoh kode program yang terdapat pada semua sub bab dalam modul ini menggunakan google colab.
- Dokumentasikan kegiatan dalam bentuk laporan praktikum sesuai template yang telah ditentukan.
- Setelah selesai mempraktekan semua materi, silakan kerjakan tugas untuk persiapan praktikum pertemuan selanjutnya.

Queue adalah kumpulan item di mana penambahan itemnya terjadi pada satu ujung yang dikenal sebagai "ekor" atau "rear", sedangkan penghapusan item terjadi pada ujung yang lain yang disebut sebagai "kepala" atau "head". Konsep queue mengikuti prinsip FIFO, singkatan dari First in First Out. Dalam kehidupan sehari-hari, queue sering diibaratkan sebagai antrian, di mana orang yang tiba lebih dulu akan dilayani lebih dulu. Konsep ini berbeda dengan konsep stack. Meskipun demikian, seperti stack, queue juga memiliki beberapa operasi yang dapat dilakukan, yaitu :

a. Enqueue adalah proses penambahan sebuah item ke dalam queue (antrian). Ketika antrian sudah mencapai kapasitas penuh, keadaannya disebut sebagai Overflow – Waktu yang dibutuhkan untuk operasi ini adalah O(1).
b. Dequeue adalah langkah menghapus satu item dari antrian. Item-item tersebut dihapus dalam urutan yang sama seperti saat mereka dimasukkan. Jika antrian kosong, kondisinya disebut sebagai Underflow – Waktu yang dibutuhkan untuk operasi ini adalah O(1).
c. Front digunakan untuk mengambil item yang berada di bagian depan antrian – Waktu yang dibutuhkan untuk operasi ini adalah O(1).
d. Rear digunakan untuk mengambil item terakhir dari antrian – Waktu yang dibutuhkan untuk operasi ini adalah O(1).

Sama halnya dengan stack, queue dapat diimplementasikan dengan 4 cara, yaitu :

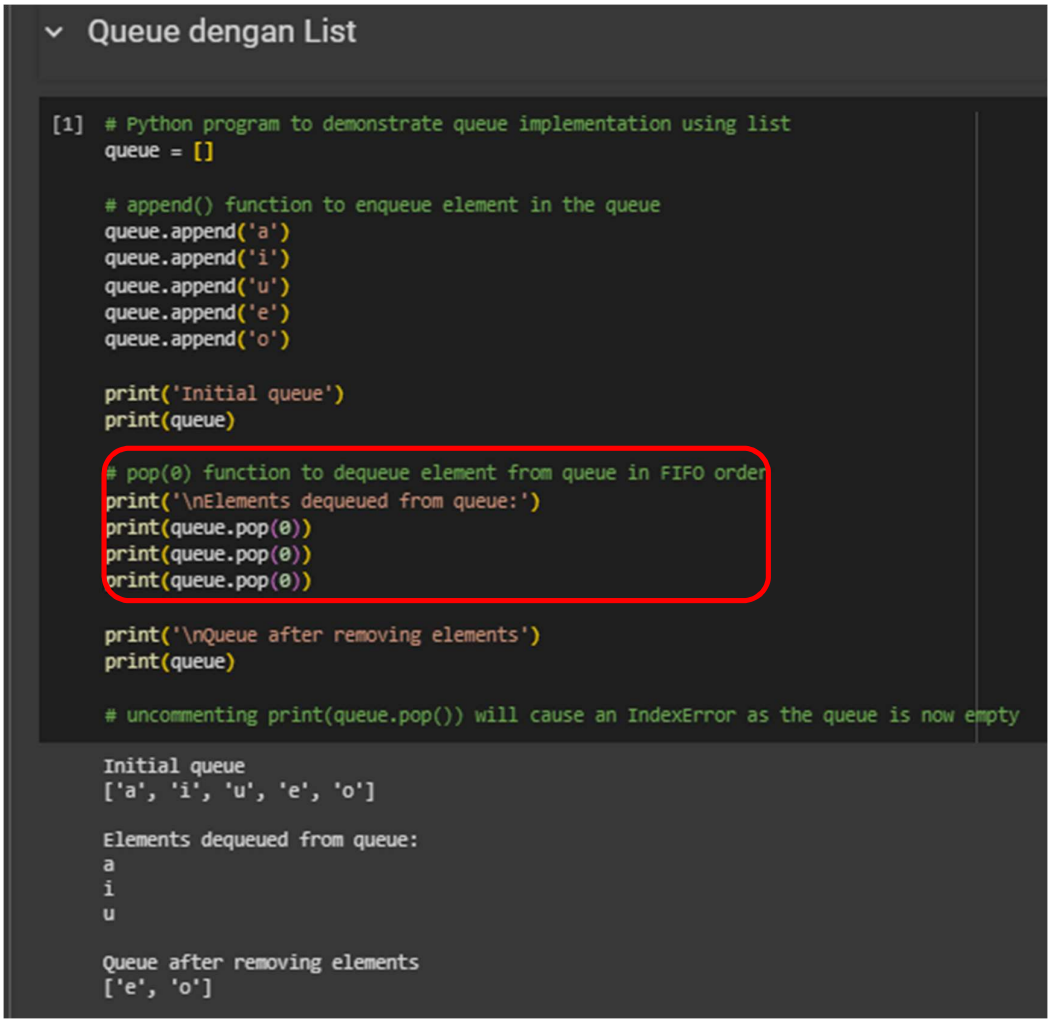
a. List
b. Collections.deque
c. Queue.Queue
d. Single Linked List

3.1 Queue dengan List

List merupakan salah satu struktur data bawaan dalam Python yang bisa berperan sebagai antrian. Sebagai gantinya, kita menggunakan fungsi append() dan pop() untuk menambah dan menghapus elemen, bukan menggunakan enqueue() dan dequeue(). Meskipun demikian, penggunaan list untuk tujuan antrian tergolong lambat karena untuk menyisipkan atau menghapus elemen di awal, semua elemen lainnya harus digeser satu per satu, yang membutuhkan waktu O(n).

Kode dibawah ini mensimulasikan penggunaan antrian menggunakan list dalam Python. Pertama-tama, elemen ' a', 'i', ‘u', 'e', dan 'o' ditambahkan ke dalam antrian, kemudian dieksekusi proses dequeue, sehingga menyebabkan antrian menjadi kosong. Output dari program ini menampilkan antrian awal, elemen-elemen yang di-dequeue ('a', 'i', 'u'), dan keadaan antrian yang kosong pada akhirnya.

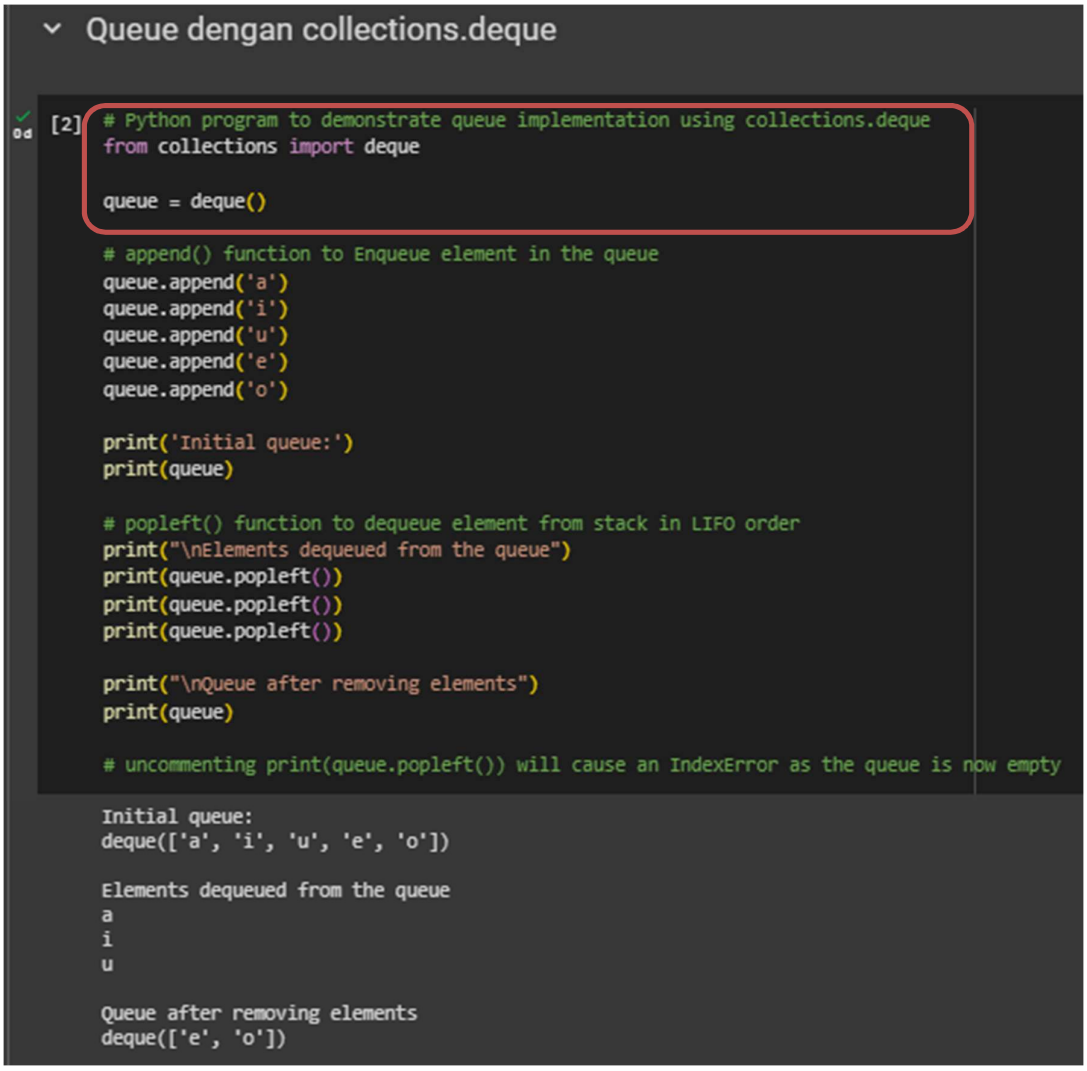
Perhatikan perbedaan dengan stack pada modul sebelumnya.



3.2 Queue dengan Collections.deque

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

Kode menggunakan deque dari modul collections untuk merepresentasikan sebuah antrian. Ini menambahkan ' a', 'i', ‘u', 'e', dan 'o' ke dalam antrian dan mengeluarkannya dengan q.popleft(), menyebabkan antrian menjadi kosong. Menghilangkan komentar dari q.popleft() setelah antrian kosong akan menyebabkan IndexError. Kode tersebut menunjukkan operasi-operasi antrian dan menangani skenario antrian kosong.



3.3 Queue dengan queue.LifoQueue

Queue adalah bagian dari modul standar Python yang berfungsi untuk menerapkan struktur data antrian. Dengan menggunakan queue.Queue(maxsize), kita dapat menginisialisasi sebuah variabel yang memiliki batas maksimum jumlah elemen yang dapat ditampung (maxsize). Ketika maxsize diberi nilai nol '0', itu berarti antrian tidak memiliki batasan jumlah elemen yang dapat dimasukkan. Prinsip yang diikuti oleh antrian ini adalah FIFO (First In First Out), yang berarti item yang pertama dimasukkan akan menjadi yang pertama dikeluarkan.

Modul ini menyediakan sejumlah fungsi yaitu :

a. maxsize - Menunjukkan jumlah maksimum item yang diizinkan dalam antrian.
b. empty() - Mengembalikan nilai True jika antrian kosong, dan False sebaliknya.
c. full() - Mengembalikan nilai True jika antrian sudah mencapai batas maksimum (maxsize). Jika antrian diinisialisasi dengan maxsize=0 (default), maka fungsi full() tidak akan pernah mengembalikan nilai True.
d. get() - Menghapus dan mengembalikan sebuah item dari antrian. Jika antrian kosong, proses akan menunggu hingga ada item yang tersedia.
e. get_nowait() - Mengembalikan sebuah item jika langsung tersedia, jika tidak, maka akan menimbulkan error (QueueEmpty).
f. put(item) - Memasukkan sebuah item ke dalam antrian. Jika antrian sudah penuh, proses akan menunggu hingga ada slot kosong tersedia sebelum menambahkan item baru.
g. put_nowait(item) - Memasukkan sebuah item ke dalam antrian tanpa memblokir. Jika tidak ada slot kosong yang langsung tersedia, akan menimbulkan error (QueueFull).
h. qsize() - Mengembalikan jumlah item yang saat ini ada dalam antrian.

Sebagai contoh, kode ini menggunakan kelas Queue dari modul antrian. Dimulai dengan sebuah antrian kosong, kemudian mengisinya dengan ' a', 'i', ‘u', 'e', dan 'o'. Setelah proses dequeue, antrian menjadi kosong, namun kemudian ditambahkan '1'. Meskipun sebelumnya kosong, antrian tetap dianggap penuh karena ukuran maksimumnya telah ditetapkan menjadi 5. Kode tersebut memperlihatkan operasi-operasi yang dapat dilakukan pada antrian, termasuk pemeriksaan untuk kondisi kosong atau penuh.

Queue dengan queue.Queue

```python
# Python program to demonstrate queue implementation using queue module
from queue import Queue

# Initializing a queue
qu = Queue(maxsize=5)

# qsize() show the number of elements in the queue
print(qu.qsize())

# put() function to enqueue element in the queue
qu.put('a')
qu.put('i')
qu.put('u')
qu.put('e')
qu.put('o')

print("Full: ", qu.full())
print("Size: ", qu.qsize())

# get() function to dequeue element from queue
print("\nElements dequeued from the queue")
print(qu.get())
print(qu.get())
print(qu.get())

print("\nEmpty: ", qu.empty())
print("Full: ", qu.full())
```

```
0
Full:  True
Size:  5

Elements dequeued from the queue
a
i
u

Empty:  False
Full:  False
```

3.4 Queue dengan Single Linked List

Antrian adalah sesuatu yang kita sering temui dalam kehidupan sehari-hari kita ketika kita menunggu untuk mendapatkan layanan. Konsep struktur data antrian memiliki makna yang sama, di mana elemen data diatur dalam urutan tertentu. Keistimewaan dari antrian terletak pada proses penambahan dan penghapusan elemen. Elemen-elemen diperbolehkan masuk dari satu sisi antrian, tetapi dihapus dari sisi lainnya. Oleh karena itu, ini merupakan prinsip First-in-First-out. Dalam menerapkan sebuah antrian menggunakan single linked list dalam Python metode insert() dan pop() digunakan untuk menambahkan dan menghapus elemen. Tidak ada proses penyisipan karena elemen data selalu ditambahkan di ujung antrian.

a. Menambahkan Elemen

Penambahan elemen dengan membuat sebuah kelas antrian mengikuti metode First-in-First-out. Metode insert bawaan digunakan untuk menambahkan elemen data.

b. Menghapus Elemen

Penerapana mengahpus elemen dengan membuat sebuah kelas antrian seperti contoh dibawa ini, di mana setelah memasukkan data kemudian menghapus data menggunakan metode pop bawaan.

TUGAS 4

Rangkumlah tentang algoritma pencarian dan pengurutan dalam python !

Tugas dikumpulkan pada elita maksimal 1 jam sebelum pertemuan selanjutnya.