Penerapan Queue pada Python

Queue adalah struktur data yang mengikuti prinsip FIFO (First-In-First-Out), dimana elemen yang pertama masuk akan menjadi yang pertama keluar. Python menyediakan modul “queue" yang menyediakan implementasi queue yang mudah digunakan. Beberapa penerapan queue dalam Python termasuk:

1. Queue Sederhana (Simple Queue):

Implementasi paling dasar dari queue menggunakan modul “queue.Queue”.

Contoh penggunaan:

from queue import Queue

# Membuat objek queue

my\_queue = Queue()

# Menambahkan elemen ke dalam queue

my\_queue.put(1)

my\_queue.put(2)

my\_queue.put(3)

# Mengambil dan menghapus elemen dari queue

first\_item = my\_queue.get()

1. Priority Queue:

Queue prioritas mengurutkan elemen berdasarkan prioritas. Modul “queue.PriorityQueue” digunakan untuk implementasi ini. Contoh:

from queue import PriorityQueue

# Membuat objek priority queue

my\_priority\_queue = PriorityQueue()

# Menambahkan elemen dengan prioritas

my\_priority\_queue.put((1, "Prioritas Tinggi"))

my\_priority\_queue.put((3, "Prioritas Rendah"))

my\_priority\_queue.put((2, "Prioritas Sedang"))

# Mengambil dan menghapus elemen dari priority queue

highest\_priority\_item = my\_priority\_queue.get()

1. Deque (Double-Ended Queue):

“collection.deque” dapat digunakan untuk membuat deque, yang mirip dengan deque tetapi memungkinkan penambahan dan penghapusan elemen di kedua ujungnya. Contoh:

from collections import deque

# Membuat objek deque

my\_deque = deque()

# Menambahkan elemen ke deque

my\_deque.append(1)

my\_deque.appendleft(2)

# Menghapus dan mengambil elemen dari deque

last\_item = my\_deque.pop()

first\_item = my\_deque.popleft()

Penerapan queue sangat berguna dalam berbagai scenario, seperti pengelolaan tugas (task) pada sistem antrian, algoritma pencarian (Breadth-First Search), dan situasi lain dimana urutan pemrosesan data adalah factor kritis