**Laporan Week 12 Stack & Queue**

*class StackQueue:*

*def \_\_init\_\_(self):*

*self.stack = []*

*self.queue = []*

*def push(self, item):*

*self.stack.append(item)*

*print(f"Item {item} ditambahkan ke stack.")*

*def pop(self):*

*if self.is\_stack\_empty():*

*print("Stack kosong, tidak dapat pop.")*

*return None*

*return self.stack.pop()*

*def peek\_stack(self):*

*if self.is\_stack\_empty():*

*print("Stack kosong.")*

*return None*

*return self.stack[-1]*

*def is\_stack\_empty(self):*

*return len(self.stack) == 0*

*def enqueue(self, item):*

*self.queue.append(item)*

*print(f"Item {item} ditambahkan ke queue.")*

*def dequeue(self):*

*if self.is\_queue\_empty():*

*print("Queue kosong, tidak dapat dequeue.")*

*return None*

*return self.queue.pop(0)*

*def peek\_queue(self):*

*if self.is\_queue\_empty():*

*print("Queue kosong.")*

*return None*

*return self.queue[0]*

*def is\_queue\_empty(self):*

*return len(self.queue) == 0*

*sq = StackQueue()*

*sq.push(10)*

*sq.push(20)*

*sq.push(30)*

*print(f"Pop dari stack: {sq.pop()}")*

*print(f"Peek stack: {sq.peek\_stack()}")*

*sq.enqueue(100)*

*sq.enqueue(200)*

*sq.enqueue(300)*

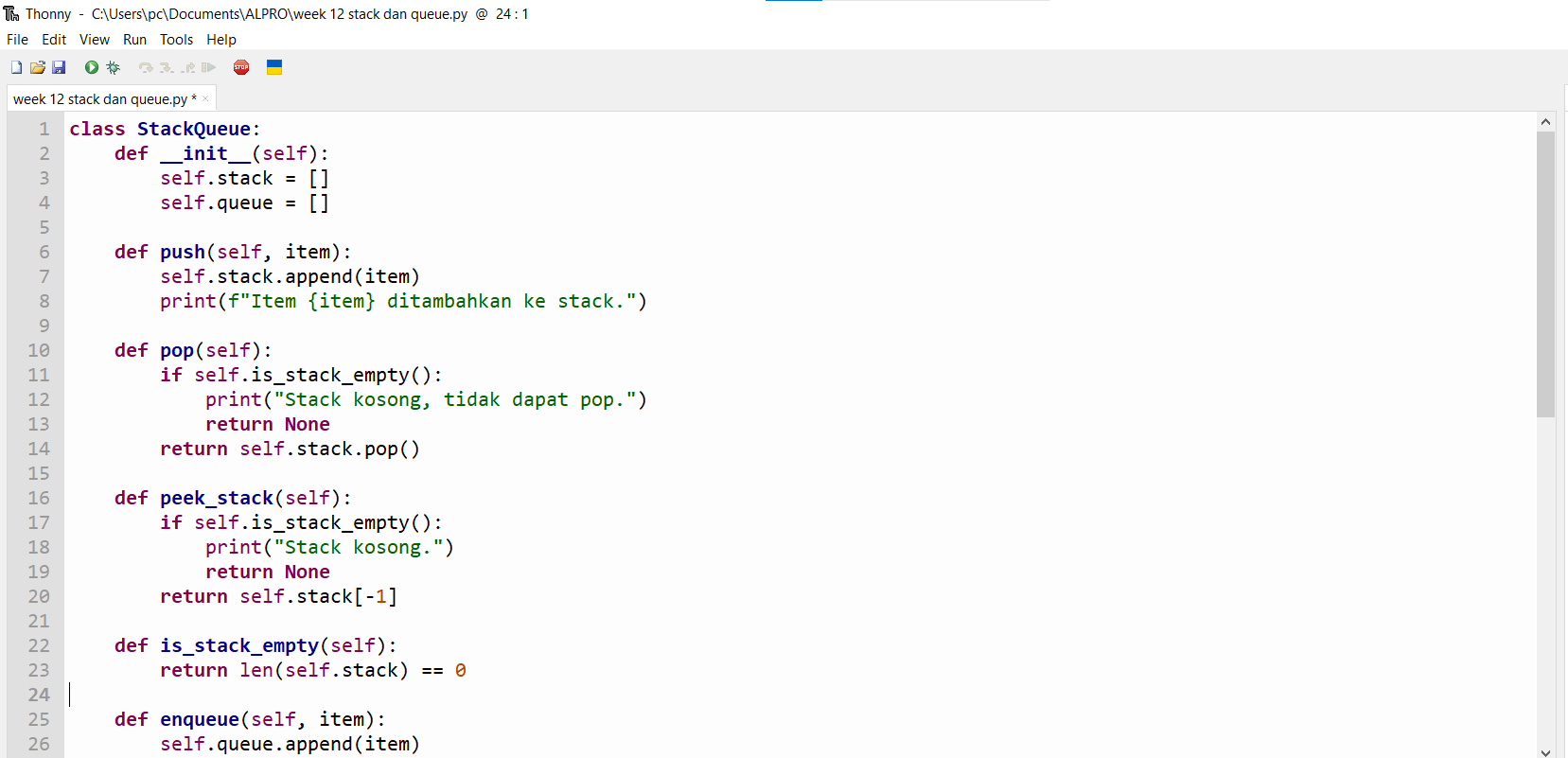
*print(f"Dequeue dari queue: {sq.dequeue()}")*

*print(f"Peek queue: {sq.peek\_queue()}")*

**Tujuan atau Hasil Akhir Program**

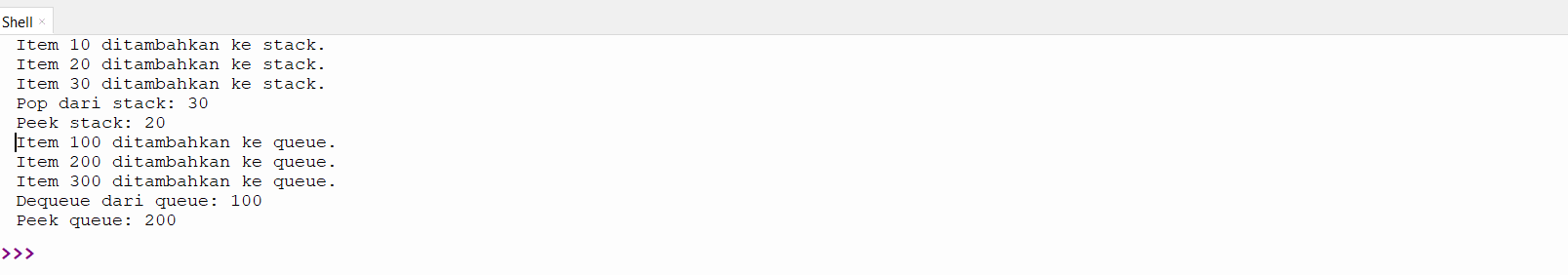
**TUJUAN**

Program ini ditujukan untuk mengimplementasikan dua struktur data dasar, yaitu Stack dan Queue, dalam 1 kelas yang sama. Kedua struktur data tersebut memiliki metode operasi yang berbeda, tapi program ini memungkinkan pengguna untuk melakukan operasi push/pop pada Stack dan enqueue/dequeue pada Queue secara bersamaan dalam satu program.



**HASIL PROGRAM**

Program berhasil mengimplementasikan operasi stack dan queue pada 1 program



1. Stack :

Push : menambah block/elemen ke stack

Pop : mengeluarkan block/elemen dari stack (prinsip last in, first out)

Peek stack : melihat elemen/block atas tanpa menghapus element tersebut

Is stack empty : melihat apakah stack kosong

1. Queue :

Enqueue : menambah elemen/antrian dalam queue

Dequeue : mengeluarkan elemen dari queue (first in, first out)

Peek queue : melihat queue pertama tanpa menghapusnya

Is queue empty : mengecek queue kosong