

Kelas `Node`

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
```

Kelas `Node` merepresentasikan elemen tunggal dalam stack.

Atribut `data` menyimpan nilai dari node.

Atribut `next` menunjuk ke node berikutnya (menghubungkan node satu dengan lainnya).

Kelas `Stack`

```
class Stack:
    def __init__(self, capacity):
        self.top = None
        self.capacity = capacity
        self.size = 0
```

top`: menunjuk ke elemen paling atas (terbaru) dalam stack.

capacity`: kapasitas maksimum stack.

size`: jumlah elemen saat ini dalam stack.

Fungsi `push(self, data)`

```
def push(self, data):
    if self.is_full():
        print("Stack sudah penuh!")
        return
    new_node = Node(data)
    new_node.next = self.top
    self.top = new_node
    self.size += 1
    print(f"Stack: {self.display()}")
```

Menambahkan elemen baru ke atas stack jika belum penuh.

Menghubungkan node baru ke node sebelumnya.

Fungsi `pop(self)`

```
def pop(self):
    if self.is_empty():
        print("Stack kosong!")
        return None
    popped_node = self.top
```

```
self.top = self.top.next
self.size -= 1
return popped_node.data
```

Menghapus elemen paling atas dari stack.
Mengembalikan nilai yang dihapus.

Fungsi `is_empty(self)`

```
def is_empty(self):
    return self.size == 0
```

- Mengecek apakah stack kosong.

Fungsi `is_full(self)`

```
def is_full(self):
    return self.size == self.capacity
```

Mengecek apakah stack penuh.

Fungsi `peek(self)`

```
def peek(self):
    if self.is_empty():
        print("Stack kosong!")
        return None
    return self.top.data
```

Mengembalikan nilai puncak stack tanpa menghapusnya.

Fungsi `display(self)`

```
def display(self):
    current = self.top
    elements = []
    while current:
        elements.append(current.data)
        current = current.next
    elements.reverse()
    return elements
```

Menampilkan seluruh isi stack dari bawah ke atas.

Fungsi ``get_size(self)``

```
def get_size(self):  
    return self.size
```

Mengembalikan ukuran stack saat ini.

Fungsi ``main()``

Menjalankan program utama untuk berinteraksi dengan pengguna melalui menu:

1. Menambah isi stack
2. Menghapus isi stack
3. Cek ukuran stack
4. Lihat puncak stack
5. Cek apakah stack penuh
6. Keluar dari program

Logika alur utama berbasis input pengguna dan perulangan ``while True``.