

Circular Stack : Structure Data

Federico Matthew Pratama - 233405001

Full Code :

```
class CircularStack:
    def __init__(self, size):
        self.size = size
        self.stack = [None] * size
        self.top = - 1
        self.count = 0

    def is_full(self):
        return self.count == self.size

    def is_empty(self):
        return self.count == 0

    def push(self, data):
        if self.is_full():
            print("Stack is Full")
            return
        self.top = (self.top + 1) % self.size
        self.stack[self.top] = data
        self.count += 1
        print(f"Added data : {data}")

    def pop(self):
        if self.is_empty():
            print("Stack is Empty")
            return
        popped = self.stack[self.top]
        self.stack[self.top] = None
        self.top = (self.top - 1 + self.size) % self.size
        self.count -= 1
        print(f"Popped data : {popped}")
        return popped

    def display_stack(self):
        print(f"Stack : {self.stack}")

    def display_count(self):
        print(f"Count : {self.count}")

if __name__ == "__main__":
    size = int(input("Masukkan ukuran queue : "))
```

```

cs = CircularStack(size)

while True:
    print("===== MENU =====")
    print("1. Push (Add)")
    print("2. Pop (Delete)")
    print("3. Tampilkan Stack")
    print("4. Tampilkan Count")
    print("5. Keluar")

    pilihan = input("Pilih menu : ")

    if pilihan == "1":
        data = int(input("Masukkan data : "))
        cs.push(data)

    elif pilihan == "2":
        cs.pop()

    elif pilihan == "3":
        cs.display_stack()

    elif pilihan == "4":
        cs.display_count()

    elif pilihan == "5":
        print("Tuhan Memberkati.")
        break

    else:
        print("Invalid Menu")

```

Pertama deklarasi variable (Menggunakan OOP -> Opsional)

- Size untuk ukuran data
- Stack untuk menyimpan size dalam list
- Top untuk start dari data kosong
- Count untuk menghitung keseluruhan Data

```

class CircularStack:
    def __init__(self, size):
        self.size = size
        self.stack = [None] * size
        self.top = - 1
        self.count = 0

```

Membuat fungsi pengecekan apabila penuh atau kosong (opsional)

```

def is_full(self):
    return self.count == self.size

```

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

Membuat fungsi untuk Push (Add Data)

- self.is_full() untuk cek data kalau penuh, dan return kalau penuh
- self.top untuk menambahkan var top (secara melingkar)
- self.stack[self.top] untuk menyimpan data
- self.count untuk menambah jumlah data

```
def push(self, data):  
    if self.is_full():  
        print("Stack is Full")  
        return  
    self.top = (self.top + 1) % self.size  
    self.stack[self.top] = data  
    self.count += 1  
    print(f"Added data : {data}")
```

Membuat fungsi Pop (Delete Data)

- self.is_empty untuk cek apakah data kosong, return kalau kosong
- membuat variable popped untuk mengambil data popped (terdahulu)
- self.top untuk mengurangi var top (secara melingkar)
- self.count untuk mengurangi jumlah data
- return popped ini digunakan untuk mengambil data yang baru masuk, agar si pop tidak bingung data mana yang di pop, anggap data baru masuk 5, maka popped ini menyimpan si 5.

```
def pop(self):  
    if self.is_empty():  
        print("Stack is Empty")  
        return  
    popped = self.stack[self.top]  
    self.stack[self.top] = None  
    self.top = (self.top - 1 + self.size) % self.size  
    self.count -= 1  
    print(f"Popped data : {popped}")  
    return popped
```

Fungsi untuk menampilkan Stack dan Count

```
def display_stack(self):  
    print(f"Stack : {self.stack}")  
  
def display_count(self):  
    print(f"Count : {self.count}")
```

Main

- size untuk menginput berapa ukuran data
- cs untuk mengambil data size dari class CircularStack
- Setelah itu hanya pemanggilan fungsi

```
if __name__ == "__main__":
    size = int(input("Masukkan ukuran queue : "))
    cs = CircularStack(size)

    while True:
        print("===== MENU =====")
        print("1. Push (Add)")
        print("2. Pop (Delete)")
        print("3. Tampilkan Stack")
        print("4. Tampilkan Count")
        print("5. Keluar")

        pilihan = input("Pilih menu : ")

        if pilihan == "1":
            data = int(input("Masukkan data : "))
            cs.push(data)

        elif pilihan == "2":
            cs.pop()

        elif pilihan == "3":
            cs.display_stack()

        elif pilihan == "4":
            cs.display_count()

        elif pilihan == "5":
            print("Tuhan Memberkati.")
            break

        else:
            print("Invalid Menu")
```

Dokumentasi

Size Data 5 :

```
Masukkan ukuran queue : 5
===== MENU =====
1. Push (Add)
2. Pop (Delete)
3. Tampilkan Stack
4. Tampilkan Count
5. Keluar
Pilih menu : █
```

Add Data sampai Full :

```
Pilih menu : 1
Masukkan data : 1
Added data : 1
```

```
Pilih menu : 1
Masukkan data : 2
Added data : 2
```

```
Pilih menu : 1
Masukkan data : 3
Added data : 3
```

```
Pilih menu : 1
Masukkan data : 4
Added data : 4
```

```
Pilih menu : 1
Masukkan data : 5
Added data : 5
```

Kalau sudah Full :

```
Masukkan data : 6
Stack is Full
```

Cek isi Stack :

```
Pilih menu : 3
Stack : [1, 2, 3, 4, 5]
```

Cek isi Jumlah :

```
Pilih menu : 4  
Count : 5
```

Pop Data :

```
Pilih menu : 2  
Popped data : 5
```

Cek Stack setelah Pop Data :

```
Pilih menu : 3  
Stack : [1, 2, 3, 4, None]
```

Cek Jumlah setelah Pop Data :

```
Pilih menu : 4  
Count : 4
```

Kalau Pop pada saat Data kosong :

```
Pilih menu : 2  
Stack is Empty
```

Keluar dari Program

```
Pilih menu : 5  
Tuhan Memberkati.
```