

# Laporan Tugas Individu: Implementasi Stack dan Queue dalam Python

AbdullahAzzam Rabbani

10240038

Universitas Bina Sarana Informatika Jl. Margonda Raya No.8, Pondok Cina, Kecamatan Beji, Kota Depok, Jawa Barat

## 1. Pendahuluan

Stack (tumpukan) dan Queue (antrian) adalah struktur data linier yang fundamental dalam ilmu komputer.

- **Stack** mengikuti prinsip LIFO (*Last In, First Out*), di mana elemen terakhir yang dimasukkan akan menjadi elemen pertama yang dikeluarkan.
- **Queue** mengikuti prinsip FIFO (*First In, First Out*), di mana elemen pertama yang dimasukkan akan menjadi elemen pertama yang dikeluarkan.

Tugas ini bertujuan untuk mengimplementasikan kedua struktur data tersebut menggunakan Python.

## 2. Implementasi Stack

Source Code:

```
class Stack:
    def __init__(self):
        self.items = []

    def push(self, item):
        self.items.append(item)

    def pop(self):
        if not self.is_empty():
            return self.items.pop()
        return "Stack kosong"

    def is_empty(self):
        return len(self.items) == 0

    def peek(self):
        if not self.is_empty():
            return self.items[-1]
        return "Stack kosong"

# Contoh Penggunaan
```

```
s = Stack()
s.push(1)
s.push(2)
s.push(3)
print(s.pop()) # Output: 3
print(s.pop()) # Output: 2
print(s.peek()) # Output: 1
print(s.pop()) # Output: 1
print(s.pop()) # Output: Stack kosong
```

### Penjelasan:

- `push(item)`: Menambahkan elemen ke bagian atas stack.
- `pop()`: Menghapus dan mengembalikan elemen teratas. Jika kosong, mengembalikan pesan.
- `peek()`: Mengembalikan elemen teratas tanpa menghapusnya.
- `is_empty()`: Memeriksa apakah stack kosong.

### Hasil Running Program:

```
D:\MatkulDasarPem\strukturData\StruckturData
3
2
1
1
Stack kosong
```

## 3. Implementasi Queue

### Source Code:

```
class Queue:
    def __init__(self):
        self.items = []

    def enqueue(self, item):
        self.items.append(item)

    def dequeue(self):
        if not self.is_empty():
            return self.items.pop(0)
        return "Queue kosong"

    def is_empty(self):
        return len(self.items) == 0

    def peek(self):
```

```

        if not self.is_empty():
            return self.items[0]
        return "Queue kosong"

# Contoh Penggunaan
q = Queue()
q.enqueue('a')
q.enqueue('b')
q.enqueue('c')
print(q.dequeue()) # Output: a
print(q.dequeue()) # Output: b
print(q.peek())    # Output: c
print(q.dequeue()) # Output: c
print(q.dequeue()) # Output: Queue kosong

```

### Penjelasan:

- `enqueue(item)`: Menambahkan elemen ke akhir antrian.
- `dequeue()`: Menghapus dan mengembalikan elemen depan. Jika kosong, mengembalikan pesan.
- `peek()`: Mengembalikan elemen depan tanpa menghapusnya.
- `is_empty()`: Memeriksa apakah antrian kosong.

### Hasil Running Program:

```

D:\MatkulDasarPem\strukturData\StruckturData
a
b
c
c
Queue kosong

Process finished with exit code 0

```

## 4. Kesimpulan

- Stack dan Queue berhasil diimplementasikan menggunakan list di Python.
- Stack menggunakan `append()` dan `pop()` untuk operasi LIFO.
- Queue menggunakan `append()` dan `pop(0)` untuk operasi FIFO.

- Kode program telah diuji dan menghasilkan output sesuai prinsip masing-masing struktur data.