

Nama : Andi Muh. Ibnu Hibban Bagoes Malolo

NIM : H071211060

Kelas : Struktur Data B

## **STACK DAN QUEUE**

### **A. Stack/Tumpukan**

Stack adalah struktur data linier yang mengikuti aturan tertentu untuk melakukan operasi. Data yang memiliki struktur stack, tersusun seperti tumpukan, sehingga hanya elemen yang baru dimasukkan yang dapat diakses atau dilihat. Stack merupakan perintah pengumpulan data secara linear yang menyusun data seperti tumpukan dan bersifat Last In First Out (LIFO) yang berarti data yang terakhir masuk adalah data yang pertama keluar. Contoh dalam Kehidupan sehari-hari :

- Setumpuk koran, dimana koran yang paling terakhir ditambahkan dan ditaruh diatas adalah koran yang dapat dilihat
- Seseorang yang sedang mencuci piring, piring yang dicuci pertama pasti akan diletakan dibawah dan akan terus berlanjut sampai tumpukan piring yang terakhir dicuci. Lalu, piring pasti akan di taruh di rak piring dan pasti yang diambil adalah piring yang paling atas yaitu piring yang terakhir dicuci dan yang pertama dicuci pasti akan terakhir.
- Tumpukan Batu bata yang sedang diturunkan dari mobil pasti yang diambil adalah batu bata yang paling atas, padahal batu bata pertama yang dimasukkan kedalam mobil adalah batu bata yang berada dibawahnya.

Jadi, Stack yaitu sebuah koleksi objek yang menggunakan prinsip LIFO (Last In First Out) dan Stack dapat diimplementasikan sebagai representasi berkait atau kontiniu.

Ciri-Ciri Stack diantaranya :

1. Elemen TOP/Puncak diketahui
2. Penyisipan dan Penghapusan selalu dilakukan di TOP
3. LIFO (Last In First Out)

Pemanfaatan Stack yaitu :

1. Perhitungan Ekspresi Aritmatika (posfix)
2. Algoritma Backtracking (runut balik)
3. Algoritma Rekursif

Operasi Stack yang biasa digunakan diantaranya yaitu :

1. Push untuk memasukkan atau menginputkan data
2. Pop untuk mengeluarkan data
3. IsFull untuk mengetahui jika tumpukan sudah penuh
4. IsEmpty untuk mengetahui tumpukan yang kosong, dan
5. Clear untuk menghapus seluruh data atau membersihkan data.

Contoh program stack dalam pemrograman Java:

```
import java.lang.reflect.Array;

public class Stack <T> {
    public T[] array;
    public int pointer;
    public Stack(Class<?> classname){
        array=(T[]) Array.newInstance(classname, 10);

        pointer = -1;
    }

    public void push(T item){
        array[++pointer]=item;
    }
    public T pop(){
        return array[pointer--];
    }

    public static void main(String[] args){
        Stack<Integer> a = new Stack<Integer>(Integer.class);
        a.push(5);
        a.push(6);
        a.push(7);

        System.out.println(a.pop());
        System.out.println(a.pop());
    }
}
```

Outputnya



```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL Run: Stack + - [] 
PS C:\Users\LENOVO\Documents\Tugas Struktur Data> c:: cd 'c:\Users\LENOVO\Documents\Tugas Struktur Data'; & 'C:\Program
Files\Java\jdk-19\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\LENOVO\AppData
ta\Roaming\Code\User\workspaceStorage\5fb150ca7a574eb56145857fd4d430c8\redhat.java\jdt_ws\Tugas Struktur Data_4cbe9b04\bi
n' 'Stack'
7
6
PS C:\Users\LENOVO\Documents\Tugas Struktur Data> 
```

## B. Queue/Antrian

Queue adalah struktur data linier di mana kita dapat menyisipkan dan menghapus elemen dari daftar data. Akhir daftar dari mana elemen disisipkan disebut ujung belakang dan ujung di mana elemen dihapus adalah ujung depan. Struktur data yang menggunakan queue mengikuti prinsip FIFO (First In First Out), yang berarti elemen yang dimasukkan pertama kali dari ujung belakang akan menjadi elemen pertama yang dihapus dari ujung depan. Contohnya saat mengantri dalam loket untuk membeli tiket. Istilah Enqueue cukup sering dipakai seseorang ketika masuk antrian. Yang datang terlebih pertama, maka akan dilayaniterlebih dahulu. Dan istilah untuk seseorang keluar dari antrian adalah Dequeue.

- Queue mempunyai beberapa fungsi operasi diantaranya yaitu :
- EnQueue untuk Memasukkan data kedalam Antrian
- DeQueue untuk Mengeluarkan data kedalam Antrian.
- IsFull untuk memeriksa apakah antrian Penuh
- IsEmpety untuk memeriksa apakah antrian Kosong
- Clear untuk Menghapus seluruh Antrian.

Dalam kehidupan sehari-hari, ada banyak sekali tentang Queue atau antrian. Contohnya adalah sebagai berikut :

1. Saat seseorang mengantri di sebuah Bank
2. Antrian Loket pembelian sebuah tiket Pesawat, Kereta Api, dan lainnya
3. Pembayaran Tol dan sebagainya.

Contoh dalam Pembelian Tiket Kereta Api :

- Enqueue :Seseorang membeli tiket melalui tempat pembayaran tiket yang disediakan.
- Dequeue : Setelah membeli tiket, langsung menuju tempat tunggu Kereta Api, dengan sebelumnya petugas memeriksa cek tiket tersebut.
- Clear : Pembeli tiket tersebut telah terhapus dari antrian karena sudah melewati pembayaran administrasi tersebut.
- IsEmpty : Petugas tiket Kereta Api melihat tidak ada lagi yang ingin membeli tiket kereta.
- IsFull : Petugas Tiket Kereta Api melihat masih ada pembeli tiket kereta.

Contoh program queue dalam pemrograman Java.

```
import java.lang.reflect.Array;
import java.util.Scanner;
public class Queue<T> {
    public Integer pointer;
    private Integer sizeArray;
    private T[] element;

    public Queue(Class<?> namaClass, Integer size) {
        pointer = -1;
```

```

        sizeArray = size;
        element = (T[]) Array.newInstance(namaClass, size);
    }

    public void enqueue(T data)
    {
        if(pointer < (sizeArray - 1))
        {
            pointer++;
            element[pointer] = data;
        }
        else
        {
            System.out.println("Queue Sudah Penuh");
        }
    }

    public void dequeue()
    {
        if(pointer >= 0) {
            for (Integer i = 0; i <= pointer; i++) {
                element[i] = element[i + 1];
            }
            pointer--;
        }
        else
        {
            System.out.println("Queue Sudah Kosong");
        }
    }

    public void viewQueue()
    {
        for(Integer i = 0; i <= pointer; i++)
        {
            System.out.print(element[i]);
            System.out.print(" ");
        }
    }

    public void clearQueue()
    {
        pointer = -1;
    }

    public static void main(String[] args)

```

```

{
    Integer pil;
    Queue <String> queue = new Queue<>(String.class, 10);
    Scanner scanner = new Scanner(System.in);

    do
    {
        queue.viewQueue();
        System.out.println();
        System.out.println("1. Enqueue");
        System.out.println("2. Dequeue");
        System.out.println("3. Clear");
        System.out.println("4. Exit");
        pil = scanner.nextInt();
        if(pil > 0 && pil < 4)
        {
            if (pil == 1)
            {
                String data;
                System.out.print("Masukkan Data: ");
                data = scanner.next();
                queue.enqueue(data);
            } else if (pil == 2)
                queue.dequeue();
            else if (pil == 3)
                queue.clearQueue();
        }
        else
            break;
    }while(true);
}
}

```

Outputnya

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL
Run: Queue + - []

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\LENOVO\Documents\Tugas Struktur Data> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '--enable-preview' '-XX:
+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\LENOVO\AppData\Roaming\Code\User\workspaceStorage\5fb150ca7a574eb561
45857fd4d430c8\redhat.java\jdt_ws\Tugas Struktur Data_4cbe9b04\bin' 'Queue'

1. Enqueue
2. Dequeue
3. Clear
4. Exit
█

```

### C. Perbedaan Stack dan Queue

Terdapat beberapa perbedaan antara stack dan queue. Berikut beberapa perbedaannya.

#### **Stack**

- Stack Didasarkan pada prinsip LIFO (Last In First Out).
- Terdapat operasi penyisipan (operasi push) dan operasi penghapusan (operasi pop).
- Operasi push dan pop berlangsung dari salah satu ujung tumpukan data.
- Elemen paling atas adalah elemen yang paling mudah diakses, sementara elemen yang berada pada tumpukan paling bawah adalah elemen yang tidak dapat diakses.
- Penerapannya sederhana.

#### **Queue**

- Didasarkan pada prinsip FIFO (First In First Out)
- Terdapat teknik penyisipan (operasi enqueue) dan teknik penghapusan (operasi dequeue) Operasi enqueue dan dequeue berlangsung dari ujung antrian yang berbeda
- Ujung penyisipan disebut dengan ujung belakang, sementara ujung penghapusan disebut ujung depan
- Penerapannya lebih kompleks dibanding dengan struktur data stack