

LAPORAN PRAKTIKUM STRUKTUR DATA

Modul ke : 2
Judul Praktikum : Stack
Hari dan Tanggal Pelaksanaan : Rabu, 19 April 2022
Tempat Pelaksanaan : Lab Desain
Dosen Pengampu Praktikum : Khoirul Umam, S.Pd, M.Kom

Nama Mahasiswa Pelaksana : Andri WIjaksono
NIM Pelaksana : 362155401206

A. Tugas Pendahuluan

Tuliskan hasil pengerjaan Tugas Pendahuluan pada bagian ini.

1. Jelaskan konsep stack!
2. Sebutkan dan jelaskan dua operasi dasar pada stack!

B. Kegiatan Praktikum

Cantumkan apa saja yang dilakukan pada latihan-latihan praktikum, *source code* yang dipakai, *screen shot* hasil eksekusi kode, dan jawaban dari pertanyaan-pertanyaan yang muncul pada tiap kegiatan latihan.

Latihan 1: Implementasi stack menggunakan array

1. Buat sebuah file bernama Stack.java kemudian tuliskan deklarasi interface Stack berikut ke dalam file tersebut. Interface Stack ini akan memastikan setiap class yang mengimplementasikannya harus mendeklarasikan ulang method-method Stack.

```
public interface Stack<T> {  
    abstract boolean isEmpty();  
    abstract T peek();  
    abstract T pop();  
    abstract void push(T item);  
    abstract int size();  
}
```

2. Pada folder yang sama dengan interface Stack tersebut buat kembali sebuah file bernama **StackArray.java** kemudian tuliskan kode berikut:

```
import java.util.EmptyStackException;  
public class StackArray<T> implements Stack<T> {  
    T value[]; // array yang digunakan sebagai stack  
    int topOfStack; // penanda posisi ruang kosong stack  
    // constructor  
    public StackArray(int size) {  
        value = (T[]) new Object[size]; // inisialisasi ukuran stack  
    }  
    @Override  
    public boolean isEmpty() {  
        // Stack kosong jika penanda ruang kosongnya ada di indeks 0  
        return topOfStack == 0;  
    }  
}
```

```

@Override
public T pop() {
    if (isEmpty())
        throw new EmptyStackException();
    // Turunkan penanda ruang kosong satu tingkat lalu
    // ambil item pada posisi penanda tersebut
    topOfStack--;
    return value[topOfStack];
}

@Override
public void push(T item) {
    // Simpan item yang dimasukkan ke ruang dimana penanda berada
    // lalu naikkan penanda satu tingkat
    value[topOfStack] = item;
    topOfStack++;
}

@Override
public T peek() {
    if (isEmpty())
        throw new EmptyStackException();
    // Item puncak (peek) merupakan item yang persis berada
    // di bawah penanda
    topOfStack--;
    T temp = value[topOfStack];
    topOfStack++;
    return temp;
}

@Override
public int size() {
    // Ukuran stack sama dengan indeks penanda berada
    return topOfStack;
}

public void printStack() {
    System.out.println("-----");
    System.out.println("Isi stack");
    System.out.println("-----");
    for (int i = topOfStack - 1; i >= 0; i--) {
        System.out.println(i + ": " + value[i]);
    }
}
}

```

3. Untuk menguji class di atas, buat kembali file baru bernama **TestStackArray.java** di folder yang sama dengan dua file sebelumnya lalu tuliskan kode berikut:

```

public class TestStackArray {
    public static void main(String[] args) {
        StackArray<String> sa = new StackArray<String>(10);
        sa.push("Pink");
    }
}

```

```

sa.push("Purple");
sa.push("Red");
sa.printStackTrace();
System.out.println("Ukuran stack : " + sa.size());
String popped = sa.pop();
System.out.println("Item yang dikeluarkan : " + popped);
System.out.println("Item puncak : " + sa.peek());
sa.printStackTrace();
System.out.println("Ukuran stack : " + sa.size());
}
}

```

```

PS C:\Users\ASUS\Documents\Struktur Data\modul2> javac TestStackArray.java
Note: .\StackArray.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
PS C:\Users\ASUS\Documents\Struktur Data\modul2> java TestStackArray
-----
Isi stack
-----
2: Red
1: Purple
0: Pink
Ukuran stack : 3
Item yang dikeluarkan : Red
Item puncak : Purple
-----
Isi stack
-----
1: Purple
0: Pink
Ukuran stack : 2

```

5. Dari hasil yang didapatkan, dimana posisi item pertama (Pink) yang dimasukkan ke stack disimpan? Dimana pula posisi item terakhir (Red) yang dimasukkan ke stack disimpan?

Jawab: Ketika memasukan data kedalam memori memiliki urutannya masing masing artinya data yang paling pertama kita push (masukan) akan berada pada bagian terbawah, dan data terakhir yang dimasukan akan berada diposisi paling atas. Pada program yang kita buat urutan push masing masing data adalah Pink ke purple lalu Red jadi bisa diambil kesimpulan pink berada pada posisi paling bawah dan Red berada pada posisi paling atas.

6. Item mana yang dikeluarkan pada saat prosedur pop pertama kali dijalankan?

Jawab: Red dikarenakan item red masuk paling akhir jadi posisinya berada di atas disitulah mengapa item red yang akan dikeluarkan terlebih dahulu,

7. Apa yang akan terjadi jika prosedur pop kembali dijalankan?

Jawab: Maka item teratas dari stack yaitu Red akan dikeluarkan, seperti halnya output yang kita lihat dengan hasil "Item yang dikeluarkan: Red".

Latihan 2: Implementasi stack menggunakan ArrayList

1. Di folder yang sama dengan file-file latihan sebelumnya, buat kembali sebuah file baru bernama **StackArrayList.java** dan tuliskan kode berikut ke dalamnya:

```

import java.util.ArrayList;
import java.util.EmptyStackException;
import java.util.ListIterator;
public class StackArrayList<T> implements Stack<T> {
    private ArrayList<T> stackList = null;
    public StackArrayList() {
        stackList = new ArrayList<T>();
    }
    @Override
    public boolean isEmpty() {
        return stackList.size() == 0;
    }
    @Override
    public T pop() {
        if (isEmpty()) {
            throw new EmptyStackException();
        }
        return stackList.remove(stackList.size() - 1);
    }
    @Override
    public void push(T item) {
        stackList.add(item);
    }
    @Override
    public T peek() {
        if (isEmpty()) {
            throw new EmptyStackException();
        }
        return stackList.get(stackList.size() - 1);
    }
    @Override
    public int size() {
        return stackList.size();
    }
    public void printStack() {
        System.out.println("-----");
        System.out.println("Isi stack");
        System.out.println("-----");
        ListIterator<T> li = stackList.listIterator(size());
        while (li.hasPrevious()) {
            System.out.println(li.previous());
        }
    }
}

```

2. Tambahkan pula file baru bernama TestStackArrayList.java di folder yang sama dan tuliskan kode berikut:

```

public class TestStackArrayList {

```

```

public static void main(String[] args) {
StackArrayList<String> sal = new StackArrayList<String>();
sal.push("Pink");
sal.push("Purple");
sal.push("Red");
sal.printStack();
System.out.println("Ukuran stack : " + sal.size());
String popped = sal.pop();
System.out.println("Item yang dikeluarkan : " + popped);
System.out.println("Item puncak : " + sal.peek());
sal.printStack();
System.out.println("Ukuran stack : " + sal.size());
}
}

```

```

-----
Isi stack
-----
Red
Purple
Pink
Ukuran stack : 3
Item yang dikeluarkan : Red
Item puncak : Purple
-----
Isi stack
-----
Purple
Pink
Ukuran stack : 2

```

4. Mengapa digunakan class ListIterator untuk menampilkan susunan item-item di dalam stack?

Jawab: Karena untuk menampilkan sebuah data dari objek atau dari kasus ini adalah stack menggunakan method List() digunakan untuk menampilkan suatu data dari stack dengan posisi tertentu. Kesimpulannya listIterator digunakan karena kita ingin keluaran data dari stacklist dengan posisi tertentu.

Latihan 3: Implementasi stack menggunakan Java Stack Collection

1. Buat file baru bernama **StackExample.java** kemudian tuliskan kode berikut:

```

import java.util.Stack;
import java.util.ListIterator;
public class StackExample {
public static void main(String[] args) {
Stack<String> s = new Stack<String>();
s.push("Pink");
s.push("Purple");
s.push("Red");
printStack(s);
}
}

```

```

System.out.println("Item puncak : " + s.peek());
System.out.println("Ukuran stack : " + s.size());
System.out.println("Ambil item : " + s.pop());
System.out.println("Ukuran stack : " + s.size());
printStats(s);
System.out.println("Cari item Red : " + s.search("Red"));
System.out.println("Cari item Purple : " + s.search("Purple"));
System.out.println("Cari item Pink : " + s.search("Pink"));
System.out.println("Apakah stack kosong? : " + s.empty());
}
public static void printStack(Stack<?> s) {
System.out.println("-----");
System.out.println("Isi stack");
System.out.println("-----");
ListIterator<?> li = s.listIterator(s.size());
while (li.hasPrevious()) {
System.out.println(li.previous());
}
}
}
}

```

```

-----
Isi stack
-----
Red
Purple
Pink
Item puncak : Red
Ukuran stack : 3
Ambil item : Red
Ukuran stack : 2
-----
Isi stack
-----
Purple
Pink
Cari item Red : -1
Cari item Purple : 1
Cari item Pink : 2
Apakah stack kosong? : false

```

3. Bagaimana hasil pencarian data/item “Red”, “Purple”, dan “Pink”? Apa makna tiap bilangan yang dihasilkan?

Jawab: Pada isi stack 3 yang pertama merupakan urutan dari posisi item dalam sebuah objek tersebut. Red diatas Purple ditengah, dan Pink dibawah setelah itu sistem melaksanakan pop pada item Red maka isi dari stack tersebut dan urutannya berubah menjadi 2 isi dan item yang tersisa hanyalah Purple dan Pink begitu juga dengan urutannya berubah Purple menjadi diatas, Pink menjadi ditengah.

C. Tugas Praktikum

Tuliskan dan jabarkan hasil pengerjaan Tugas Praktikum yang tertera di dalam modul lengkap dengan *source code* yang digunakan.

4. Tugas Praktikum

Buatlah aplikasi sederhana yang menerapkan konsep stack untuk:

1. Konversi dari nilai desimal ke biner, oktal, dan heksadesimal.

```
import java.util.Scanner;

import java.util.Scanner;

class Converter {
    Scanner scan;
    int num;
    void getVal()
    {
        scan = new Scanner(System.in);
        System.out.println("Masukan Angka :");
        num = Integer.parseInt(scan.nextLine());
    }
    void convert()
    {
        String hexa = Integer.toHexString(num);
        System.out.println("Nilai Bilangan Heksadesimal : " + hexa);
        String octal = Integer.toOctalString(num);
        System.out.println("Nilai Bilangan Oktal : " + octal);
        String binary = Integer.toBinaryString(num);
        System.out.println("Nilai Bilangan Biner : " + binary);
    }
}

class Decimal_Conversion {
    public static void main(String args[]) {
        Converter obj = new Converter();
        obj.getVal();
        obj.convert();
    }
}
```

```
PS C:\Users\ASUS\Documents\Struktur Data\modul2> javac Converter.java
PS C:\Users\ASUS\Documents\Struktur Data\modul2> java Decimal_Conversion
Masukan Angka :
31
Nilai Bilangan Heksadesimal : 1f
Nilai Bilangan Oktal : 37
Nilai Bilangan Biner : 11111
```

```

PS C:\Users\ASUS\Documents\Struktur Data\modul2> javac Converter.java
PS C:\Users\ASUS\Documents\Struktur Data\modul2> java Decimal_Conversion
Masukan Angka :
25
Nilai Bilangan Heksadesimal : 19
Nilai Bilangan Oktal : 31
Nilai Bilangan Biner : 11001

```

2. Memeriksa apakah sebuah kalimat atau string termasuk palindrom atau bukan.

```

import java.util.*;

class PalindromeChecker {
    public static void main(String args[]) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Pemeriksa Palindrom");
        System.out.println("-----");
        System.out.print("Masukan String yang ingin diperiksa: ");
        String str = sc.nextLine();
        String reverse = "";

        for(int i=str.length()-1;i>=0;i--){
            reverse = reverse + str.charAt(i);
        }
        if (str.equals(reverse)){
            System.out.println("String '"+str+"' Kata ini adalah Palindrome");
        } else {
            System.out.println("String '"+str+"' Ini bukan kata Palindrome!");
        }
    }
}

```

```

Pemeriksa Palindrom
-----
Masukan String yang ingin diperiksa: struktur data
String 'struktur data' Ini bukan kata Palindrome!
PS C:\Users\ASUS\Documents\Struktur Data> c:: cd 'c:\Use
\Java\jdk-16.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExc
Code\User\workspaceStorage\c78a9db63e778c5e9392fbef2683b8
lindromeChecker'
Pemeriksa Palindrom
-----
Masukan String yang ingin diperiksa: kasur ini rusak
String 'kasur ini rusak' Kata ini adalah Palindrome
PS C:\Users\ASUS\Documents\Struktur Data> 

```