JOBSHEET VII STACK

**Nama : M Abhinaya Z  
Kelas : SIB 1 C / 20**

# Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

* + 1. Mengenal struktur data Stack
    2. Membuat dan mendeklarasikan struktur data Stack
    3. Menerapkan algoritma Stack dengan menggunakan array

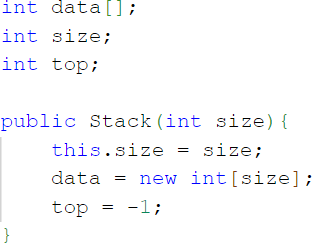
# Praktikum 1

## Waktu percobaan: 30 menit

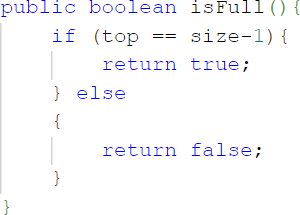
Pada percobaan ini, kita akan membuat program yang mengimplementasikan struktur data Stack dan operasi-operasi dasar pada struktur data Stack menggunakan array.

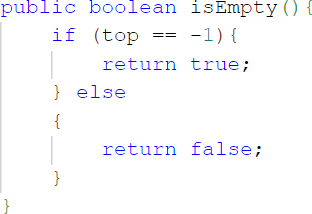
# Langkah-langkah Percobaan

* + - 1. Buat folder dengan nama Praktikum07. Buat file Stack.java.
      2. Tulis kode untuk membuat atribut dan konstruktor pada class Stack sebagai berikut:

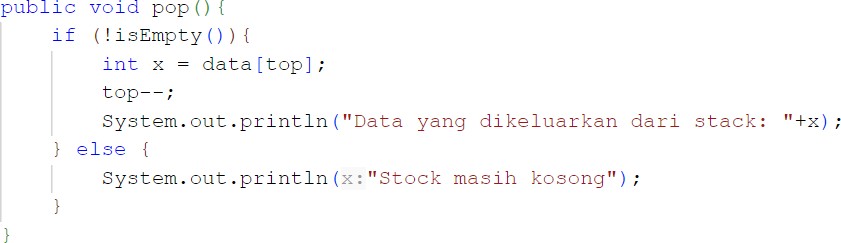
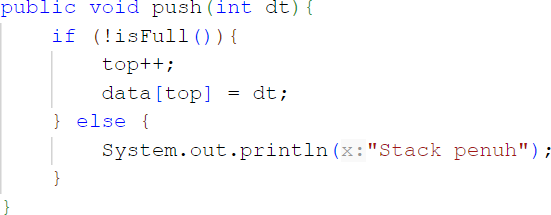


* + - 1. Lalu tambahkan method isFull() dan isEmpty() pada class Stack sebagai berikut:

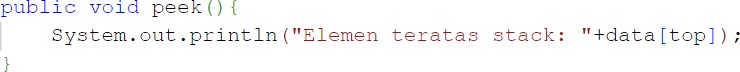




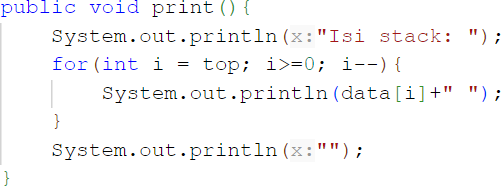
* + - 1. Tambahkan method push(int data) dan pop() sebagai berikut:

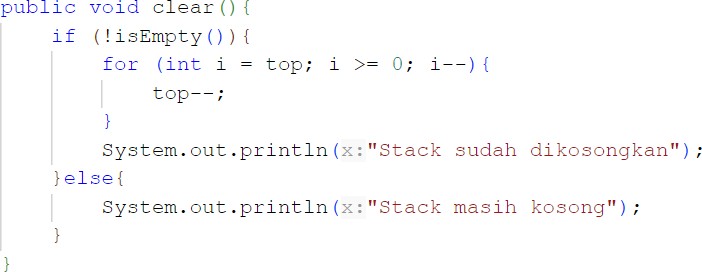


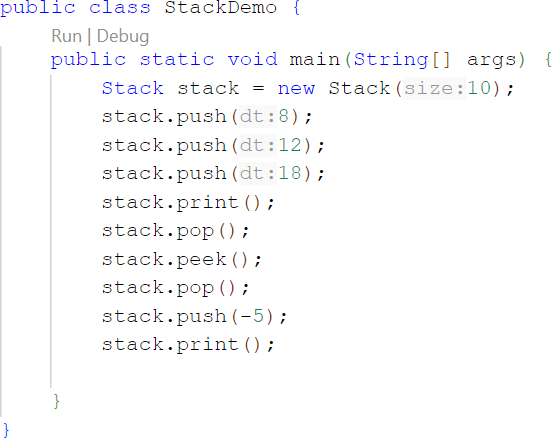
* + - 1. Tambahkan method peek() sebagai berikut:



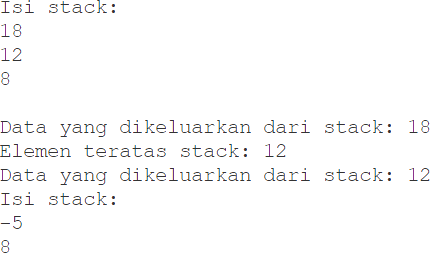
* + - 1. Tambahkan method print() dan clear() sebagai berikut:





* + - 1. Buat file **StackDemo.java** untuk mengimplementasikan class StackDemo yang berisi fungsi main untuk membuat objek Stack dan mengoperasikan method-method pada class Stack.
      2. Compile dan run class StackDemo.

# Verifikasi Hasil Percobaan



* + 1. **Pertanyaan**
       1. Pada method pop(), mengapa diperlukan pemanggilan method isEmpty()? Apa yang terjadi jika tidak ada pemanggilan isEmpty()?

Diperlukan pemanggilan method isEmpty() untuk memeriksa apakah stack masih kosong sebelum menghapus top of stack, jika tidak ada pemanggilan isEmpty maka yang terjadi akan menyebabkan error karena mencoba mengambil elemen dari stack yang tidak ada / indeks yang tidak valid.

* + - 1. Jelaskan perbedaan antara method peek() dengan method pop() pada class Stack.
* Method peek() hanya berfungsi untuk mengakses elemen top of stack dan pada kode tersebut menampilkan elemen tersebut
* Method pop() berfungsi untuk mengambil/mengeluarkan elemen top of stack dan juga mengembalikan nilai yang diambil

# Praktikum 2

## Waktu percobaan : 45 menit

Pada percobaan ini, kita akan membuat program yang mengilustrasikan tumpukan pakaian yang disimpan ke dalam stack. Karena sebuah pakaian mempunyai beberapa informasi, maka implementasi Stack dilakukan dengan menggunakan array of object untuk mewakili setiap elemennya.

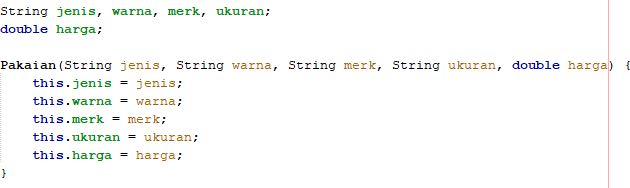
# Langkah-langkah Percobaan

1. Perhatikan Diagram Class Pakaian berikut ini:

|  |
| --- |
| Pakaian |
| jenis: String warna: String merk: String ukuran: String  harga: double |
| Pakaian(jenis: String, warna: String, merk: String, ukuran: String, harga: double) |

Berdasarkan diagram class tersebut, akan dibuat program class Pakaian dalam Java.

1. Buat class baru dengan nama **Pakaian**.
2. Tambahkan atribut-atribut Pakaian seperti pada Class Diagram Pakaian, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.

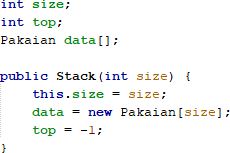


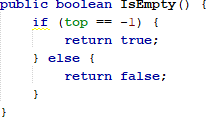
1. Setelah membuat class Pakaian, selanjutnya perlu dibuat class **Stack** yang berisi atribut dan method sesuai diagram Class Stack berikut ini:

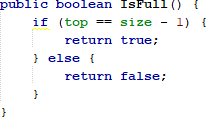
|  |
| --- |
| Stack |
| size: int top: int  data[]: Pakaian |
| Stack(size: int) IsEmpty(): boolean IsFull(): boolean  push(): void |

|  |
| --- |
| pop(): void peek(): void print(): void  clear(): void |

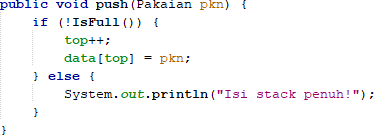
***Keterangan****: Tipe data pada variabell* ***data*** *menyesuaikan dengan data yang akan akan disimpan di dalam Stack. Pada praktikum ini, data yang akan disimpan merupakan array of object dari Pakaian, sehingga tipe data yang digunakan adalah* ***Pakaian***

1. Buat class baru dengan nama Stack. Kemudian tambahkan atribut dan konstruktor seperti gambar berikut ini.
2. Buat method **IsEmpty** bertipe boolean yang digunakan untuk mengecek apakah stack kosong.

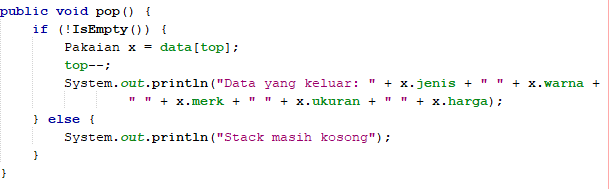


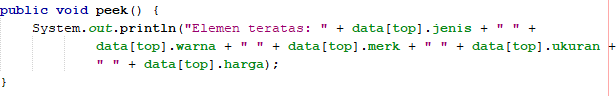
1. Buat method **IsFull** bertipe boolean yang digunakan untuk mengecek apakah stack sudah terisi penuh.
2. Buat method **push** bertipe void untuk menambahkan isi elemen stack dengan parameter **pkn**

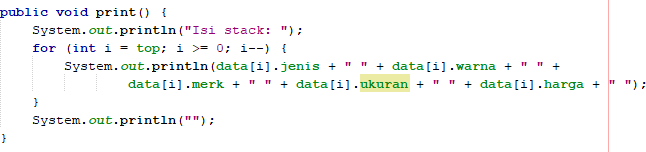
yang berupa object **Pakaian**



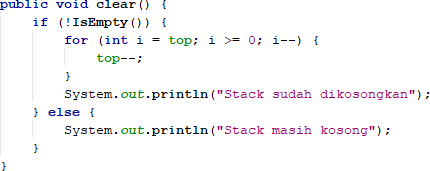
1. Buat method **Pop** bertipe void untuk mengeluarkan isi elemen stack. Karena satu elemen stack terdiri dari beberapa informasi (jenis, warna, merk, ukuran, dan harga), maka ketika mencetak data juga perlu ditampilkan semua informasi tersebut



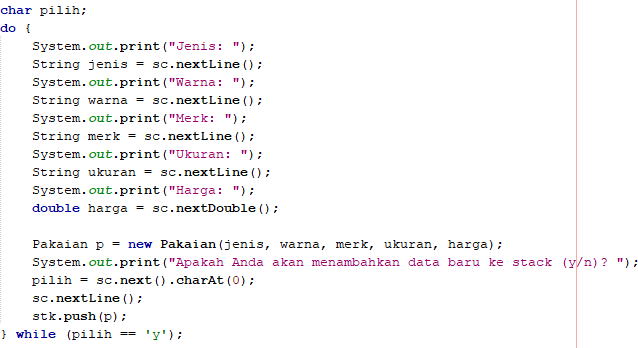
1. Buat method **peek** bertipe void untuk memeriksa elemen stack pada posisi paling atas.
2. Buat method **print** bertipe void untuk menampilkan seluruh elemen pada stack.



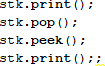
1. Buat method **clear** bertipe void untuk menghapus seluruh isi stack.



1. Selanjutnya, buat class baru dengan nama **StackMain**. Buat fungsi main, kemudian lakukan instansiasi objek dari class **Stack** dengan nama **stk** dan nilai parameternya adalah 5.
2. Deklarasikan Scanner dengan nama **sc**
3. Tambahkan kode berikut ini untuk menerima input data Pakaian, kemudian semua informasi tersebut dimasukkan ke dalam stack



***Catatan****: sintaks sc.nextLine() sebelum sintaks st.push(p) digunakan untuk mengabaikan karakter new line*

1. Lakukan pemanggilan method print, method pop, dan method peek dengan urutan sebagai berikut.
2. Compile dan jalankan class **StackMain**, kemudian amati hasilnya.

# Verifikasi Hasil Percobaan

* + 1. **Pertanyaan**

1. Berapa banyak data pakaian yang dapat ditampung di dalam stack? Tunjukkan potongan kode program untuk mendukung jawaban Anda tersebut!

Ada 5 elemen

1. Perhatikan class **StackMain**, pada saat memanggil fungsi push, parameter yang dikirimkan adalah

**p**. Data apa yang tersimpan pada variabel **p** tersebut?

Data yang tersimpan pada p adalah data-data/atribut yang di input oleh user melalui scanner

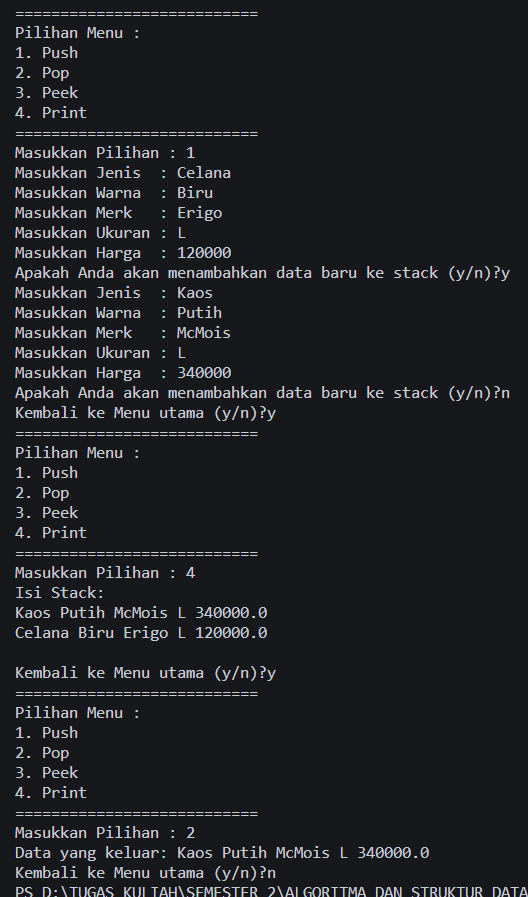
1. Apakah fungsi penggunaan **do-while** yang terdapat pada class **StackMain**?

Berfungsi untuk melakukan perulangan untuk user dapat menginput data yang berisi detail pakaian, Ketika setelah menginput, user akan diberi pilihan

"Apakah Anda akan menambahkan data baru ke stack (y/n)? "

Ketika “y” maka akan melakukan perulangan dan user akan menginput detail lagi dan sampai jika user memilih “n” maka perulangan akan berhenti dan program selesai.

1. Modifikasi kode program pada class **StackMain** sehingga pengguna dapat memilih operasi- operasi pada stack (push, pop, peek, atau print) melalui pilihan menu program dengan memanfaatkan kondisi IF-ELSE atau SWITCH-CASE!



package praktikum07;

*/\*\**

*\* @author Abhinaya*

*\*/*

import java.util.Scanner;

public class StackMain {

    public static void main(String[] args) {

        Stack20 stk = new Stack20(5);

        Scanner sc = new Scanner(System.in);

        char pilih;

        do {

            System.out.println("===========================");

            System.out.println("Pilihan Menu : ");

            System.out.println("1. Push");

            System.out.println("2. Pop");

            System.out.println("3. Peek");

            System.out.println("4. Print");

            System.out.println("===========================");

            System.out.print("Masukkan Pilihan : ");

            int pilihan = sc.nextInt();

            switch (pilihan) {

                case 1:

                    do {

                        System.out.print("Masukkan Jenis\t: ");

                        String jenis = sc.next();

                        System.out.print("Masukkan Warna\t: ");

                        String warna = sc.next();

                        System.out.print("Masukkan Merk\t: ");

                        String merk = sc.next();

                        System.out.print("Masukkan Ukuran\t: ");

                        String ukuran = sc.next();

                        System.out.print("Masukkan Harga\t: ");

                        double harga = sc.nextDouble();

                        Pakaian p = new Pakaian(jenis, warna, merk, ukuran, harga);

                        System.out.print("Apakah Anda akan menambahkan data baru ke stack (y/n)?");

                        pilih = sc.next().charAt(0);

                        stk.push(p);

                    } while (pilih == 'y');

                    break;

                case 2:

                stk.pop();

                break;

                case 3:

                stk.peek();

                break;

                case 4:

                stk.print();

                break;

                default:

                System.out.println("Pilihan Tidak Tersedia");

                break;

            } if (pilihan == 5) {

                pilih = 'n';

            } else {

                System.out.print("Kembali ke Menu utama (y/n)?");

                pilih = sc.next().charAt(0);

            }

        } while (pilih == 'y');

    }

}