



JOBSHEET VII

STACK

Nama : M Abhinaya Z

Kelas : SIB 1 C / 20

7.1 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

7.2. 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.

7.2.1 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:

```
int data[];
int size;
int top;

public Stack(int size){
    this.size = size;
    data = new int[size];
    top = -1;
}
```

3. Lalu tambahkan method isFull() dan isEmpty() pada class Stack sebagai berikut:

```
public boolean isFull(){
    if (top == size-1){
        return true;
    } else
    {
        return false;
    }
}
```

```
public boolean isEmpty(){
    if (top == -1){
        return true;
    } else
    {
        return false;
    }
}
```

4. Tambahkan method push(int data) dan pop() sebagai berikut:

```
public void push(int dt){
    if (!isFull()){
        top++;
        data[top] = dt;
    } else {
        System.out.println(x:"Stack penuh");
    }
}

public void pop(){
    if (!isEmpty()){
        int x = data[top];
        top--;
        System.out.println("Data yang dikeluarkan dari stack: "+x);
    } else {
        System.out.println(x:"Stock masih kosong");
    }
}
```

5. Tambahkan method peek() sebagai berikut:

```
public void peek(){
    System.out.println("Elemen teratas stack: "+data[top]);
}
```

6. Tambahkan method print() dan clear() sebagai berikut:

```
public void print(){
    System.out.println(x:"Isi stack: ");
    for(int i = top; i>=0; i--){
        System.out.println(data[i]+" ");
    }
    System.out.println(x:"");
}
```



```
public void clear(){
    if (!isEmpty()){
        for (int i = top; i >= 0; i--){
            top--;
        }
        System.out.println(x:"Stack sudah dikosongkan");
    }else{
        System.out.println(x:"Stack masih kosong");
    }
}
```

7. Buat file **StackDemo.java** untuk mengimplementasikan class StackDemo yang berisi fungsi main untuk membuat objek Stack dan mengoperasikan method-method pada class Stack.

```
public class StackDemo {
    Run | Debug
    public static void main(String[] args) {
        Stack stack = new Stack(size:10);
        stack.push(dt:8);
        stack.push(dt:12);
        stack.push(dt:18);
        stack.print();
        stack.pop();
        stack.peek();
        stack.pop();
        stack.push(-5);
        stack.print();
    }
}
```

8. Compile dan run class StackDemo.

7.2.2 Verifikasi Hasil Percobaan

Isi stack:

18
12
8

Data yang dikeluarkan dari stack: 18

Elemen teratas stack: 12

Data yang dikeluarkan dari stack: 12

Isi stack:

-5
8

```
PS D:\TUGAS KULIAH\SEMESTER 2\ALGORITMA
CodeDetailsInExceptionMessages' '-cp' '
Isi stack :
18
12
8
Data yang dikeluarkan dari Stack: 18
Elemen teratas stack: 12
Data yang dikeluarkan dari Stack: 12
Isi stack :
-5
8
PS D:\TUGAS KULIAH\SEMESTER 2\ALGORITMA
```

7.2.3 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.



2. 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

7.3. 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.

7.3.1. 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.

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

```
String jenis, warna, merk, ukuran;
double harga;

Pakaian(String jenis, String warna, String merk, String ukuran, double harga) {
    this.jenis = jenis;
    this.warna = warna;
    this.merk = merk;
    this.ukuran = ukuran;
    this.harga = harga;
}
```

4. 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 variabel **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**

5. Buat class baru dengan nama Stack. Kemudian tambahkan atribut dan konstruktor sepertigambar berikut ini.

```
int size;
int top;
Pakaian data[];

public Stack(int size) {
    this.size = size;
    data = new Pakaian[size];
    top = -1;
}
```

6. Buat method **IsEmpty** bertipe boolean yang digunakan untuk mengecek apakah stack kosong.

```
public boolean IsEmpty() {
    if (top == -1) {
        return true;
    } else {
        return false;
    }
}
```

7. Buat method **IsFull** bertipe boolean yang digunakan untuk mengecek apakah stack sudah terisi penuh.

```
public boolean IsFull() {
    if (top == size - 1) {
        return true;
    } else {
        return false;
    }
}
```

8. Buat method **push** bertipe void untuk menambahkan isi elemen stack dengan parameter **pkn** yang berupa object **Pakaian**

```
public void push(Pakaian pkn) {
    if (!IsFull()) {
        top++;
        data[top] = pkn;
    } else {
        System.out.println("Isi stack penuh!");
    }
}
```

9. 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

```
public void pop() {
    if (!IsEmpty()) {
        Pakaian x = data[top];
        top--;
        System.out.println("Data yang keluar: " + x.jenis + " " + x.warna +
            " " + x.merk + " " + x.ukuran + " " + x.harga);
    } else {
        System.out.println("Stack masih kosong");
    }
}
```

10. Buat method **peek** bertipe void untuk memeriksa elemen stack pada posisi paling atas.

```
public void peek() {
    System.out.println("Elemen teratas: " + data[top].jenis + " " +
        data[top].warna + " " + data[top].merk + " " + data[top].ukuran +
        " " + data[top].harga);
}
```

11. Buat method **print** bertipe void untuk menampilkan seluruh elemen pada stack.

```
public void print() {
    System.out.println("Isi stack: ");
    for (int i = top; i >= 0; i--) {
        System.out.println(data[i].jenis + " " + data[i].warna + " " +
            data[i].merk + " " + data[i].ukuran + " " + data[i].harga + " ");
    }
    System.out.println("");
}
```

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

```
public void clear() {
    if (!IsEmpty()) {
        for (int i = top; i >= 0; i--) {
            top--;
        }
        System.out.println("Stack sudah dikosongkan");
    } else {
        System.out.println("Stack masih kosong");
    }
}
```



13. 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.

```
Stack stk = new Stack(5);
```

14. Deklarasikan Scanner dengan nama **sc**

15. Tambahkan kode berikut ini untuk menerima input data Pakaian, kemudian semua informasi tersebut dimasukkan ke dalam stack

```
char pilih;
do {
    System.out.print("Jenis: ");
    String jenis = sc.nextLine();
    System.out.print("Warna: ");
    String warna = sc.nextLine();
    System.out.print("Merk: ");
    String merk = sc.nextLine();
    System.out.print("Ukuran: ");
    String ukuran = sc.nextLine();
    System.out.print("Harga: ");
    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);
    sc.nextLine();
    stk.push(p);
} while (pilih == 'y');
```

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

16. Lakukan pemanggilan method print, method pop, dan method peek dengan urutan sebagai berikut.

```
stk.print();
stk.pop();
stk.peek();
stk.print();
```

17. Compile dan jalankan class **StackMain**, kemudian amati hasilnya.



7.3.2. Verifikasi Hasil Percobaan

```
Jenis: Kaos
Warna: Hitam
Merk: Nevada
Ukuran: M
Harga: 85000
Apakah Anda akan menambahkan data baru ke stack (y/n)? y
Jenis: Kemeja
Warna: Putih
Merk: Styves
Ukuran: XL
Harga: 127000
Apakah Anda akan menambahkan data baru ke stack (y/n)? y
Jenis: Celana
Warna: Biru
Merk: Levis
Ukuran: L
Harga: 189500
Apakah Anda akan menambahkan data baru ke stack (y/n)? n
Isi stack:
Celana Biru Levis L 189500.0
Kemeja Putih Styves XL 127000.0
Kaos Hitam Nevada M 85000.0

Data yang keluar: Celana Biru Levis L 189500.0
Elemen teratas: Kemeja Putih Styves XL 127000.0
Isi stack:
Kemeja Putih Styves XL 127000.0
Kaos Hitam Nevada M 85000.0
```

```
hat.java\jdt_ws\Jobsheet 7_3f90417e\bin' 'praktikum07.Sta
Jenis: Kaos
Warna: Hitam
Merk: Nevada
Ukuran: M
Harga: 85000
Apakah Anda akan menambahkan data baru ke stack (y/n)? y
Jenis: Kemeja
Warna: Putih
Merk: Styves
Ukuran: XL
Harga: 127000
Apakah Anda akan menambahkan data baru ke stack (y/n)? y
Jenis: Celana
Warna: Biru
Merk: Levis
Ukuran: L
Harga: 189500
Apakah Anda akan menambahkan data baru ke stack (y/n)? n
Isi Stack:
Celana Biru Levis L 189500.0
Kemeja Putih Styves XL 127000.0
Kaos Hitam Nevada M 85000.0

Data yang keluar: Celana Biru Levis L 189500.0
Elemen teratas: Kemeja Putih Styves XL 127000.0
Isi Stack:
Kemeja Putih Styves XL 127000.0
Kaos Hitam Nevada M 85000.0
```

7.3.3. Pertanyaan

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

Ada 5 elemen

```
public static void main(String[] args) {
    Stack20 stk = new Stack20(size:5);
}
```

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

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

```
stk.push(p);
```

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

3. 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.

4. 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!

```
=====
Pilihan Menu :
1. Push
2. Pop
3. Peek
4. Print
=====
Masukkan Pilihan : 1
Masukkan Jenis : Celana
Masukkan Warna : Biru
Masukkan Merk : Erigo
Masukkan Ukuran : L
Masukkan Harga : 120000
Apakah Anda akan menambahkan data baru ke stack (y/n)?y
Masukkan Jenis : Kaos
Masukkan Warna : Putih
Masukkan Merk : McMois
Masukkan Ukuran : L
Masukkan Harga : 340000
Apakah Anda akan menambahkan data baru ke stack (y/n)?n
Kembali ke Menu utama (y/n)?y
=====
Pilihan Menu :
1. Push
2. Pop
3. Peek
4. Print
=====
Masukkan Pilihan : 4
Isi Stack:
Kaos Putih McMois L 340000.0
Celana Biru Erigo L 120000.0

Kembali ke Menu utama (y/n)?y
=====
Pilihan Menu :
1. Push
2. Pop
3. Peek
4. Print
=====
Masukkan Pilihan : 2
Data yang keluar: Kaos Putih McMois L 340000.0
Kembali ke Menu utama (y/n)?n
PS. D:\TUGAS KULIAH\SEMESTER 2\ALGORITMA DAN STRUKTUR DATA
```



```
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');
}
}

```