

**LAPORAN HASIL PRAKTIKUM**  
**ALGORITMA DAN STRUKTUR DATA**  
**JOBSHEET 15**



**ATHAULLA HAFIZH**

**244107020030**

**TI 1 E**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2024**

## Persiapan

### Kode program class Customer

```
package Praktikum15;

public class Customer {
    public int id;
    public String name;

    public Customer() {
    }

    public Customer(int id, String name) {
        this.id = id;
        this.name = name;
    }

    public String toString() {
        return " ID: " + this.id + " Nama: " + this.name;
    }
}
```

### Kode program class Book

```
package Praktikum15;

public class Book {
    public String isbn;
    public String title;

    public Book() {
    }

    public Book(String isbn, String title) {
```

```
        this.isbn = isbn;

        this.title = title;

    }

    public String toString() {

        return "ISBN: " + this.isbn + " Title: " + this.title;

    }

}
```

## Praktikum – Implementasi ArrayList

```
package Praktikum15;

import java.util.ArrayList;

public class DemoArrayList {

    public static void main(String[] args) {

        ArrayList<Customer> customers = new ArrayList<>(2);

        Customer customer1 = new Customer(1, "Zakia");
        Customer customer2 = new Customer(5, "Budi");

        customers.add(customer1);
        customers.add(customer2);

        customers.add(new Customer(4, "Cica"));

        customers.add(2, new Customer(100, "Rosa"));

        System.out.println(customers.indexOf(customer2));

        Customer customer = customers.get(1);
        System.out.println(customer.name);
        customer.name = "Budi Utomo";

        ArrayList<Customer> newCustomers = new ArrayList<>();
        newCustomers.add(new Customer(201, "Della"));
```

```

        newCustomers.add(new Customer(202, "Victor"));
        newCustomers.add(new Customer(203, "Sarah"));

        customers.addAll(newCustomers);

        for (Customer cust : customers) {
            System.out.println(cust.toString());
        }

        System.out.println(customers);
    }
}

```

5. Compile dan run kode program, di mana object yang baru ditambahkan? Di awal, di tengah, atau di akhir collection?

Ya, object dapat ditambahkan meskipun melebihi kapasitas. Object yang baru ditambahkan berada di akhir collection

7. Compile dan run kode program. Index pada ArrayList dimulai dari 0 atau 1?

Mulai dari 0

10. Cobalah hapus angka 2 saat instansiasi object customers. Apakah ArrayList dapat diinstansiasi tanpa harus menentukan size di awal?

Ya, ArrayList dapat diinstansiasi tanpa menentukan size di awal

Output Akhir

```

1
Budi
ID: 1 Nama: Zakia
ID: 5 Nama: Budi Utomo
ID: 100 Nama: Rosa
ID: 4 Nama: Cica
ID: 201 Nama: Della
ID: 202 Nama: Victor
ID: 203 Nama: Sarah
[ ID: 1 Nama: Zakia, ID: 5 Nama: Budi Utomo, ID: 100 Nama: Rosa, ID: 4 Nama: Cica, ID: 201 Nama: Della, ID: 202 Nama: Victor, ID: 203 Nama: Sarah]

```

## Praktikum – Implementasi TreeSet

```
package Praktikum15;

import java.util.TreeSet;

public class TreeSetDemo {
    public static void main(String[] args) {
        TreeSet<String> fruits = new TreeSet<>();

        fruits.add("Mangga");
        fruits.add("Apel");
        fruits.add("Jeruk");
        fruits.add("Jambu");

        for (String temp : fruits) {
            System.out.println(temp);
        }

        System.out.println("First: " + fruits.first());
        System.out.println("Last: " + fruits.last());

        fruits.remove("Jeruk");
        System.out.println("Setelah remove " + fruits);

        fruits.pollFirst();
        System.out.println("Setelah poll first " + fruits);

        fruits.pollLast();
        System.out.println("Setelah poll last " + fruits);
    }
}
```

## Output

```
Apel  
Jambu  
Jeruk  
Mangga  
First: Apel  
Last: Mangga  
Setelah remove [Apel, Jambu, Mangga]  
Setelah poll first [Jambu, Mangga]  
Setelah poll last [Jambu]
```

4. Compile dan run program. Mengapa urutan yang ditampilkan berbeda dengan urutan penambahan data ke dalam TreeSet fruits?

Karena karakteristik TreeSet yaitu sorted, jadi akan otomatis sesuai urutan alfabet

6. Apa yang dilakukan oleh method first(), last(), remove(), pollFirst(), dan pollLast()?

1. first()

Mengembalikan elemen pertama (terkecil) dalam TreeSet

2. last()

Mengembalikan elemen terakhir (terbesar) dalam TreeSet

3. remove(element)

Menghapus elemen tertentu dari TreeSet

Mengembalikan true jika elemen berhasil dihapus, false jika tidak ada

4. pollFirst()

Menghapus dan mengembalikan elemen pertama (terkecil)

Mengembalikan null jika TreeSet kosong (tidak melempar exception)

5. pollLast()

Menghapus dan mengembalikan elemen terakhir (terbesar)

Mengembalikan null jika TreeSet kosong (tidak melempar exception)

## Praktikum Sorting

```
package Praktikum15;

import java.util.ArrayList;
import java.util.Collections;

public class Sorting {

    public static void main(String[] args) {

        ArrayList<String> daftarSiswa = new ArrayList<>();
        daftarSiswa.add("Zainab");
        daftarSiswa.add("Andi");
        daftarSiswa.add("Rara");

        Collections.sort(daftarSiswa);
        System.out.println(daftarSiswa);

        ArrayList<Customer> customers = new ArrayList<>();
        customers.add(new Customer(3, "Zakia"));
        customers.add(new Customer(1, "Andi"));
        customers.add(new Customer(2, "Rara"));
        customers.add(new Customer(4, "Budi"));

        customers.sort((c1, c2) -> c1.name.compareTo(c2.name));
        System.out.println(customers);
    }
}
```

## Output

```
[Andi, Rara, Zainab]
[ ID: 1 Nama: Andi, ID: 4 Nama: Budi, ID: 2 Nama: Rara, ID: 3 Nama: Zakia]
```