

# **Algotimat dan Struktur Data**

## **Jobsheet 14 : Collection**



Oleh:

Fajar Eka Sandiyuda (244107020043)

Kelas TI 1F / 04

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur 65141

## Praktikum - Implementasi ArrayList

### Kode

```
public class Main {
    public static void main(String[] args) {

        customers.addAll(newCustomers);

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

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

        customers.add(new Customer(id:4, name:"Cica"));
        customers.add(index:2, new Customer(id:100, name:"Rosa"));
        System.out.println(customers.indexOf(customer2));

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

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

        for(Book book : books){
            System.out.println(book.toString());
        }

        System.out.println(books);
    }
}
```

```
1 public class Book {
2     public String isbn;
3     public String title;
4
5     public Book(){
6
7     }
8
9     public Book(String isbn, String title){
10         this.isbn = isbn;
11         this.title = title;
12     }
13
14     public String toString(){
15         return "ISBN : " + this.isbn + " Title : " + this.title;
16     }
17 }
18
```

```
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;
    }
}
```

### Pertanyaan

1. Buat file DemoArrayList.java. Lakukan import java.util.ArrayList; 2. Pada fungsi main(), instansiasi collection baru dengan nama customers bertipe ArrayList of Customer dengan size
2. Selanjutnya, buat object customer1 dan customer2 kemudian tambahkan ke dalam ArrayList customers dengan method add.
3. Gunakan looping dengan foreach untuk mencetak data customers hlm. 2 dari 9 Algoritma dan Struktur Data 2024-2025
4. Cobalah tambahkan object customer baru ke dalam customers. Apakah object dapat ditambahkan meskipun melebihi kapasitas?
5. Compile dan run kode program, di mana object yang baru ditambahkan? Di awal, di tengah, atau di akhir collection?
6. Untuk menambahkan object baru pada index tertentu, lakukan sebagai berikut
7. Compile dan run kode program. Index pada ArrayList dimulai dari 0 atau 1?
8. Untuk mengetahui posisi dari suatu objek, gunakan method indexOf()
9. Untuk mengembalikan object pada index tertentu, gunakan method get()
10. Cobalah hapus angka 2 saat instansiasi object customers. Apakah ArrayList dapat diinstansiasi tanpa harus menentukan size di awal?
11. Anda juga dapat menambahkan sekumpulan customer baru ke dalam ArrayList secara sekaligus. Misalnya terdapat ArrayList newCustomers. Tambahkan seluruh object customer sekaligus ke dalam customers.
12. Karena sudah menyediakan method toString(), pengecekan data customers untuk proses debugging juga dapat dilakukan lebih sederhana dengan cara berikut

#### Jawaban

4. Ya, ArrayList bisa menampung lebih banyak objek meski tidak ada kapasitas awal yang ditentukan. Java secara otomatis mengatur kapasitas.
5. Secara default, add() menambahkan objek di akhir.
7. Ya, ArrayList menggunakan index berbasis **0**.
10. Ya, ArrayList bisa diinstansiasi tanpa size.
12. Karena Customer sudah override toString(), kita bisa langsung mencetak objek.

## Praktikum - Implementasi Stack

### Code

```
1  import java.util.Stack;
2
3  public class BookMain {
4      Run | Debug
5      public static void main(String[] args) {
6          Stack<Book> books = new Stack<>();
7
8          Book book1 = new Book(isbn:"1213", title:"Dasar Pemrograman");
9          Book book2 = new Book(isbn:"1214", title:"Hafalan");
10         Book book3 = new Book(isbn:"1215", title:"Dayummm");
11
12         books.push(book1);
13         books.push(book2);
14         books.push(book3);
15
16         Book temp = books.peek();
17
18         if (temp != null){
19             System.out.println(temp.toString());
20         }
21
22         Book temp2 = books.pop();
23
24         if (temp2 != null){
25             System.out.println(temp2.toString());
26         }
27     }
28 }
```

### Pertanyaan

1. Buat file StackDemo.java. Lakukan import java.util.Stack;
2. Pada fungsi main() buat beberapa object bertipe Book.
3. Instansiasi object books bertipe Stack of Book kemudian tambahkan object yang sudah dibuat ke dalamnya menggunakan method push()
4. Class Stack juga sudah memiliki method pop() dan peek() seperti yang Anda diimplementasikan secara manual pada praktikum sebelumnya
5. Mengapa perlu ada pengecekan (temp != null)?
6. Lakukan looping untuk mencetak data buku pada stack
7. Jika diperlukan pada proses debugging, books juga dapat dicetak dengan cara berikut
8. Bagaimana cara melakukan pencarian elemen pada stack menggunakan method search()?

### Jawaban

8. Mengembalikan posisi dari atas (1 adalah paling atas), -1 jika tidak ditemukan.

## Praktikum - Implementasi TreeSet

### Code

```
import java.util.TreeSet;

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

        fruits.add(e:"Apple");
        fruits.add(e:"Orange");
        fruits.add(e:"Banana");
        fruits.add(e:"Mango");
        fruits.add(e:"Grapes");

        System.out.println(x:"Isi TreeSet:");
        for (String fruit : fruits) {
            System.out.println(fruit);
        }

        System.out.println("\nfirst(): " + fruits.first());
        System.out.println("last(): " + fruits.last());

        fruits.remove(o:"Mango");
        System.out.println(x:"\nSetelah remove(\"Mango\"):");
        System.out.println(fruits);

        System.out.println("\npollFirst(): " + fruits.pollFirst());
        System.out.println("pollLast(): " + fruits.pollLast());

        System.out.println(x:"\nIsi TreeSet setelah polling:");
        System.out.println(fruits);
    }
}
```

### Pertanyaan

1. Buat file TreeSetDemo.java kemudian import java.util.TreeSet;
2. Tambahkan fungsi main() kemudian instansiasi object TreeSet of String. Tambahkan beberapa nilai bertipe String ke dalam TreeSet Cetak data pada ts dengan looping compile dan run program.
3. Mengapa urutan yang ditampilkan berbeda dengan urutan penambahan data ke dalam TreeSet fruits? Tambahkan kode program sebagai berikut:
4. Apa yang dilakukan oleh method first(), last(), remove(), pollFirst(), dan pollLast()?

### Jawaban

first() : Mengembalikan elemen pertama (paling kecil)

last() : Mengembalikan elemen terakhir (paling besar)

remove(x) : Menghapus elemen tertentu dari TreeSet

`pollFirst()` : Menghapus dan mengembalikan elemen pertama

`pollLast()` : Menghapus dan mengembalikan elemen terakhir