LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
COLLECTION

AHMAD SOFYAN BADAWI

244107020473

KELAS TI-1B

PRODI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2025

Persiapan

Langkah-langkah:

1. Buat class Customer dalam Customer.java

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

    Customer() {

    }

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

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

2. Buat class Book dalam Book.java

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

    Book() {

    }

    Book(String isbn, String title) {
        this.isbn = isbn;
        this.title = title;
    }

    public String toString() {
        return "ISBN: " + this.isbn + " Title: " + this.title;
    }
}
```

Implementasi ArrayList

Langkah-langkah:

1. Buat file DemoArrayList.java. Lakukan import java.util.ArrayList;

```
import java.util.ArrayList;

public class DemoArrayList {
```

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.

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


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

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

3. Gunakan looping dengan foreach untuk mencetak data customers

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

4. Cobalah tambahkan object customer baru ke dalam customers. Apakah object dapat ditambahkan meskipun melebihi kapasitas?

```
        customers.add(new Customer(id:4, name:"Cica"));
```

5. Compile dan run kode program, di mana object yang baru ditambahkan? Di awal, di tengah, atau di akhir collection?
   Jawab: Object baru selalu ditambahkan di akhir collection
6. Untuk menambahkan object baru pada index tertentu, lakukan sebagai berikut

```
        customers.add(index:2, new Customer(id:100, name:"Rosa"));
```

7. Compile dan run kode program. Index pada ArrayList dimulai dari 0 atau 1?
   Jawab: Index pada ArrayList dimulai dari 0
8. Untuk mengetahui posisi dari suatu objek, gunakan method indexOf()

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

9. Untuk mengembalikan object pada index tertentu, gunakan method get()

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

10. Cobalah hapus angka 2 saat instansiasi object customers. Apakah ArrayList dapat diinstansiasi tanpa harus menentukan size di awal?
    Jawab: Ya, ArrayList dapat menambahkan elemen melebihi kapasitas awal karena ukurannya dinamis

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.

```
ArrayList<Customer> newCustomers = new ArrayList<>();
newCustomers.add(new Customer(id:201, name:"Della"));
newCustomers.add(new Customer(id:202, name:"Victor"));
newCustomers.add(new Customer(id:203, name:"Sarah"));

customers.addAll(newCustomers);
```

12. Karena sudah menyediakan method toString(), pengecekan data customers untuk proses debugging juga dapat dilakukan lebih sederhana dengan cara berikut

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

Implementasi ArrayList

Langkah-langkah:

1. Buat file StackDemo.java. Lakukan import java.util.Stack;

```
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()

```
Book book1 = new Book(isbn:"1234", title:"Dasar Pemrograman");
Book book2 = new Book(isbn:"7145", title:"Hafalah Shalat Delisa");
Book book3 = new Book(isbn:"3562", title:"Muhammad Al-Fatih");

Stack<Book> books = new Stack<>();
books.push(book1);
books.push(book2);
books.push(book3);
```

4. Class Stack juga sudah memiliki method pop() dan peek() seperti yang Anda diimpelementasikan secara manual pada praktikum sebelumnya

```
Book temp = books.peek();
if (temp != null) {
    System.out.println(temp.toString());
}

Book temp2 = books.pop();
if (temp2 != null) {
    System.out.println(temp2.toString());
}
```

5. Mengapa perlu ada pengecekan (temp != null)?
   Jawab: Untuk menghindari NullPointerException jika stack kosong

6. Lakukan looping untuk mencetak data buku pada stack

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

7. Jika diperlukan pada proses debugging, books juga dapat dicetak dengan cara berikut

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

8. Bagaimana cara melakukan pencarian elemen pada stack menggunakan method search()?

```
        // 8. Pencarian elemen dengan search()
        System.out.println("\nBuku 1 ada pada elemen ke-" + books.search(book1));
    }
}
```

Implementasi TreeSet

Langkah-langkah:

1. Buat file TreeSetDemo.java kemudian import java.util.TreeSet;

```
import java.util.TreeSet;
```

2. Tambahkan fungsi main() kemudian instansiasi object TreeSet of String. Tambahkan beberapa nilai bertipe String ke dalam TreeSet

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

        fruits.add(e:"Mangga");
        fruits.add(e:"Apel");
        fruits.add(e:"Jeruk");
        fruits.add(e:"Jambu");
```

3. Cetak data pada ts dengan looping

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

4. Compile dan run program. Mengapa urutan yang ditampilkan berbeda dengan urutan penambahan data ke dalam TreeSet fruits?
   Jawab: Karena TreeSet secara otomatis mengurutkan elemen berdasarkan natural ordering (untuk String: urutan alfabet)
5. Tambahkan kode program sebagai berikut:

```
        System.out.println("\nFirst: " + fruits.first());
        System.out.println("Last: " + fruits.last());

        fruits.remove(o:"Jeruk");
        System.out.println("\nSetelah remove Jeruk: " + fruits);

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

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

6. Apa yang dilakukan oleh method first(), last(), remove(), pollFirst(), dan pollLast()
   Jawab:
   - first(): Mengembalikan elemen pertama (terkecil)
   - last(): Mengembalikan elemen terakhir (terbesar)
   - remove(): Menghapus elemen tertentu
   - pollFirst(): Menghapus dan mengembalikan elemen pertama
   - pollLast(): Menghapus dan mengembalikan elemen terakhir

Implementasi Sorting

Selain pencarian data menggunakan method get() atau search(), Java Collection Framework juga menyediakan fungsi untuk melakukan pengurutan data.

Untuk melakukan pengurutan data String, int, atau primitive data type lain, Anda dapat melakukan cara berikut:

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

System.out.println(daftarSiswa);
```

Khusus untuk collection of objects, pengurutan data harus menentukan berdasarkan atribut mana pengurutan dilakukan. Misalnya ingin dilakukan pengurutan data customer pada praktikum di atas berdasarkan atribut name. Gunakan kode program berikut (berikan nama variabel c1 dan c2 secara random. Anda bisa menggunakan nama variabel lainnya)

```
customers.sort((c1, c2) -> c1.name.compareTo(c2.name));
System.out.println(x:"\nSetelah sorting berdasarkan nama:");
System.out.println(customers);
```

Kedua cara di atas dapat digunakan untuk mengurutkan data pada jenis collection lain yang tidak melakukan pengurutan secara otomatis seperti TreeSet.