

JOBSHEET COLECTION
ALGORITMA DAN STRUKTUR
DATA



Burhnauddin ihsan
244107020189
TI 1E/06

PROGRAM STUDI D-IV TEKNIK INFORMATIKA JURUSAN
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2025

PERCOBAAN 1

1. Kode program

- Customer

```
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;
    }
}
```

- Book

```
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;
    }
}
```

- DemoArrayList

```
import java.util.ArrayList;

public class DemoArrayList {
    public static void main(String[] args) {
        ArrayList<Customer> customers = new ArrayList<>(2);

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

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

        Customer customer3 = new Customer(3, "kepin");
        customers.add(customer3);

        customers.add(2, new Customer(10, "abyan"));

        System.out.println(customers.indexOf(customer3));
    }
}
```

```

Customer customer = customers.get(3);
System.out.println(customer.name);

ArrayList<Customer> newCustomers = new ArrayList<>();
newCustomers.add(new Customer(100, "Siti"));
newCustomers.add(new Customer(101, "Joko"));

customers.addAll(newCustomers);

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

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

    }
}

```

2. PERTANYAAN

4. Cobalah tambahkan object customer baru ke dalam customers. Apakah object dapat ditambahkan meskipun melebihi kapasitas?
 - bisa ditambahkan, karena itu hanyalah inisiasi awal
5. Compile dan run kode program, di mana object yang baru ditambahkan? Di awal, di tengah, atau di akhir collection?
 - di akhir
7. Compile dan run kode program. Index pada ArrayList dimulai dari 0 atau 1?
 - dari indeks 0
10. Cobalah hapus angka 2 saat instansiasi object customers. Apakah ArrayList dapat diinstansiasi tanpa harus menentukan size di awal?
 - bisa tanpa inisiasi size
11. Menambahkan seluruh object customer sekaligus

```

ArrayList<Customer> newCustomers = new ArrayList<>();
newCustomers.add(new Customer(100, "Siti"));
newCustomers.add(new Customer(101, "Joko"));

customers.addAll(newCustomers);

```

3. Hasil dari kode program

```

3
kepin
ID : 1 Nama : Budi
ID : 5 Nama : Abyan
ID : 10 Nama : abyan
ID : 3 Nama : kepin

```

- STACK

1. PERTANYAAN

5. Mengapa perlu ada pengecekan (temp != null)?
 - Karena jika datanya null saat kita pop akan error karena mengembalikan null
6. Bagaimana cara melakukan pencarian elemen pada stack menggunakan method search()?
 - Dengan cara memanggil method search pada objek yang tipe datanya arraylist stack dan parameternya diisi dengan nilai data yang mau dicari, dan nantinya akan mengembalikan index dari nilai tersebut

2. Hasil kode program

```
dt_ws\Jobsheet 14_55432b86\bin' 'StackDemo'  
ISBN : 567 Title : Pemrograman Web  
ISBN : 567 Title : Pemrograman Web  
ISBN : 123 Title : Dasar Pemrograman  
ISBN : 345 Title : Pemrograman Berbasis Objek  
Lakukan pencarian buku: 2  
[ISBN : 123 Title : Dasar Pemrograman, ISBN : 345 Title : Pemrograman Berbasis Objek]
```

- TREE SET

1. PERTANYAAN

4. Compile dan run program. Mengapa urutan yang ditampilkan berbeda dengan urutan penambahan data ke dalam TreeSet fruits?
 - Karena Tree set secara otomatis menyimpan elemen dalam urutan dari kecil ke besar (kalau abjad dari a - z). Konsepnya hampir mirip seperti binary search tree
6. Apa yang dilakukan oleh method first(), last(), remove(), pollFirst(), dan pollLast()?
 - first(): mengembalikan elemen pertama (terkecil) dalam TreeSet.
 - last() : Mengembalikan elemen terakhir (terbesar) dalam TreeSet.
 - remove(Object o) : Menghapus elemen tertentu dari TreeSet jika ada.
 - pollFirst() : Mengambil dan menghapus elemen pertama (terkecil) dari set.
 - pollLast() : Mengambil dan menghapus elemen terakhir (terbesar) dari set.

- SORTING

Hasil dari kode program

```
[ID : 5 Nama : Abyan, ID : 1 Nama : Budi, ID : 101 Nama : Joko, ID : 100 Nama : Siti, ID : 10 Nama : abyan  
, ID : 3 Nama : kepin]
```