

## User (abstract class & method)

```
User.java

package model;

public abstract class User { 2 usages 2 inheritors
    protected String email; 3 usages
    protected String username; 4 usages
    protected String password; 2 usages

    public User(String email, String username, String password) {
        this.email = email;
        this.username = username;
        this.password = password;
    }

    public String getUsername() { 3 usages 1 override
        return username; }

    public String getPassword() { 3 usages 1 override
        return password;
    }

    public abstract String getRole(); no usages 2 implementations
}
```

## Admin

```
Admin.java

package model;

public class Admin extends User{ 7 usages
    public Admin(String email, String username, String password) {
        super(email, username, password);
    }

    @Override 3 usages
    public String getUsername() {
        return super.getUsername();
    }

    @Override 3 usages
    public String getPassword() {
        return super.getPassword();
    }

    @Override
    public String toString() {
        return "Admin: " + username + ", email= " + email;
    }

    @Override no usages
    public String getRole() {
        return "Admin";
    }
}
```

## Pengguna

```
Pengguna.java

package model;

public class Pengguna extends User{ 9 usages
    public static final Pengguna DEFAULT_USER = new Pengguna( email: "user@mail.com", username: "user", password: "user123");

    public Pengguna (String email, String username, String password){ 2 usages
        super(email, username, password);
    }

    @Override
    public String toString() {
        return "Pengguna: " + username + ", email: " + email;
    }

    @Override no usages
    public String getRole() {
        return "Pengguna";
    }
}
```

## Aksesori (atribut final)

```
package model;

public class Aksesori{
    private final int id;
    public String nama;
    public String merek;
    public int harga;
    public Kategori kategori;

    public Aksesori(int id, String nama, String merek, int harga, Kategori kategori){
        this.id = id;
        this.nama = nama;
        this.merek = merek;
        this.harga = harga;
        this.kategori = kategori;
    }

    public int get_id(){
        return id;
    }
    public String get_nama(){
        return nama;
    }
    public String get_merek(){
        return merek;
    }
    public int get_harga(){
        return harga;
    }
    public Kategori get_kategori(){
        return kategori;
    }

    public void set_nama(String nama){
        this.nama = nama;
    }
    public void set_merek(String merek){
        this.merek = merek;
    }
    public void set_harga(int harga){
        this.harga = harga;
    }
    public void set_kategori(Kategori kategori){
        this.kategori = kategori;
    }

    public String dalam_string(){
        return "ID: " + id + ", Nama: " + nama + ", Merek: " + merek + ", Harga: Rp" + harga + ", Kategori: " + (kategori == null ? kategori.getNamaKategori() : "Belum ada kategori");
    }
}
```

## Kategori (class dan atribut final)

```
Kategori.java
package model;

public final class Kategori{ 11 usages
    private final int idKategori; 2 usages
    public String nama_kategori; 3 usages

    public Kategori(int idKategori, String nama_kategori){ 2 usages
        this.idKategori = idKategori;
        this.nama_kategori = nama_kategori;
    }

    public String getNamaKategori(){ 1 usage
        return nama_kategori;
    }

    public String dalam_string(){ 1 usage
        return "ID: " + idKategori + ", Nama Kategori: " + nama_kategori;
    }
}
```

## CRUD (method final)

```
CRUD.java
package service;

import java.util.ArrayList;
import java.util.Scanner;
import model.Aksesori;
import model.Kategori;

public class CRUD { 8 usages
    public static final ArrayList<Aksesori> dataAksesori = new ArrayList<>();
    public static final ArrayList<Kategori> dataKategori = new ArrayList<>();
    public static final Scanner scanner = new Scanner(System.in); 17 usages
    private static int Id = 1; 1 usage

    static{
        dataKategori.add(new Kategori( idKategori: 1, nama_kategori: "-"));
    }

    public static final void tambah_aksesori(){ 1 usage
        System.out.println("=== Tambah Aksesori ===");
        System.out.print("Nama: ");
        String nama = scanner.nextLine();
        System.out.print("Merek: ");
        String merek = scanner.nextLine();
        System.out.print("Harga: ");
        int harga = scanner.nextInt();
        scanner.nextLine();
        System.out.println("Pilih Kategori: ");
        lihat_kategori();
        System.out.print("Pilihan: ");
        int pilihan = scanner.nextInt();
        scanner.nextLine();
        Kategori kategori = dataKategori.get(pilihan - 1);
        Aksesori aksesori = new Aksesori(Id++, nama, merek, harga, kategori);
        dataAksesori.add(aksesori);
    }

    public static final void lihat_aksesori(){ 4 usages
        System.out.println("\n=== Lihat Aksesori ===");
        if (dataAksesori.isEmpty()) {
            System.out.println("Tidak ada aksesori yang tersedia.");
        } else {
            for (Aksesori aksesori : dataAksesori) {
                System.out.println(aksesori.dalam_string());
            }
        }
    }
}
```

● ● ● CRUD.java

```
public static final void ubah_aksesori(){ 1 usage
    System.out.println("=== Ubah Aksesori ===");
    lihat_aksesori();
    System.out.print("Pilih Aksesori: ");
    int pilihan = scanner.nextInt();
    scanner.nextLine();
    Aksesori aksesori = dataAksesori.get(pilihan - 1);
    System.out.print("Nama: ");
    String nama = scanner.nextLine();
    System.out.print("Merek: ");
    String merek = scanner.nextLine();
    System.out.print("Harga: ");
    int harga = scanner.nextInt();
    scanner.nextLine();
    System.out.println("Pilih Kategori: ");
    lihat_kategori();
    System.out.print("Pilihan: ");
    int pilihan_kategori = scanner.nextInt();
    scanner.nextLine();
    Kategori kategori = dataKategori.get(pilihan_kategori - 1);
    aksesori.set_nama(nama);
    aksesori.set_merek(merek);
    aksesori.set_harga(harga);
    aksesori.set_kategori(kategori);
}

public static final void hapus_aksesori(){ 1 usage
    System.out.println("=== Hapus Aksesori ===");
    lihat_aksesori();
    System.out.print("Pilih Aksesori: ");
    int pilihan = scanner.nextInt();
    scanner.nextLine();
    dataAksesori.remove(index: pilihan - 1);
}

public static final void tambah_kategori(){ 1 usage
    System.out.println("=== Tambah Kategori ===");
    System.out.print("Nama Kategori: ");
    String nama_kategori = scanner.nextLine();
    Kategori kategori = new Kategori(idKategori: dataKategori.size() + 1, nama_kategori);
    dataKategori.add(kategori);
}

public static final void lihat_kategori(){ 3 usages
    System.out.println("\n=== Lihat Kategori ===");
    if (dataKategori.isEmpty()) {
        System.out.println("Tidak ada kategori yang tersedia.");
    } else {
        for (int i = 0; i < dataKategori.size(); i++) {
            System.out.println(dataKategori.get(i).dalam_string());
        }
    }
}
}
```