Account (Interface)

User (mengimplementasi Account)

User.java

```
package model;

public abstract class User implements Account {  2 usages  2 inheritor
    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

```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

```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

Aksesori.java

```java
package model;

public class Aksesori{ 6 usages
    private final int id; 3 usages
    public String nama; 4 usages
    public String merek; 4 usages
    public int harga; 4 usages
    public Kategori kategori; 5 usages

    public Aksesori(int id, String nama, String merek, int harga, Kategori kategori){ 1 usage
        this.id = id;
        this.nama = nama;
        this.merek = merek;
        this.harga = harga;
        this.kategori = kategori;
    }

    public int get_id(){ no usages
        return id;
    }
    public String get_nama(){ no usages
        return nama;
    }
    public String get_merek(){ no usages
        return merek;
    }
    public int get_harga(){ no usages
        return harga;
    }
    public Kategori get_kategori(){ no usages
        return kategori;
    }

    public void set_nama(String nama){ 1 usage
        this.nama = nama;
    }
    public void set_merek(String merek){ 1 usage
        this.merek = merek;
    }
    public void set_harga(int harga){ 1 usage
        this.harga = harga;
    }
    public void set_kategori(Kategori kategori){ 1 usage
        this.kategori = kategori;
    }

    public String dalam_string(){ 1 usage
        return "ID: " + id + ", Nama: " + nama + ", Merek: " + merek + ", Harga: Rp" + harga + ", Kategori: " + (kategori != null ? kategori.getNamaKategori() : "Belum ada kategori");
    }
}
```

Kategori

Kategori.java

```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 (Static & Exception Handling)

CRUD.java

```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<>();  6 usages
    public static final ArrayList<Kategori> dataKategori = new ArrayList<>();  10 usages
    public static final Scanner scanner = new Scanner(System.in);  22 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();

        int harga = 0;
        while (true) {
            try {
                System.out.print("Harga: ");
                harga = scanner.nextInt();
                scanner.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Input tidak valid. Harap masukkan angka untuk harga.");
                scanner.nextLine();
            }
        }

        System.out.println("Pilih Kategori: ");
        lihat_kategori();

        int pilihan = 0;
        while (true) {
            try {
                System.out.print("Pilihan: ");
                pilihan = scanner.nextInt();
                if (pilihan < 1 || pilihan > dataKategori.size()) {
                    System.out.println("Pilihan tidak valid. Silakan pilih kategori yang tersedia.");
                    continue;
                }
                scanner.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Input tidak valid. Harap masukkan angka untuk pilihan kategori.");
                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());
            }
        }
    }

    public static final void ubah_aksesori() {  1 usage
        System.out.println("=== Ubah Aksesori ===");
        lihat_aksesori();

        int pilihan = 0;
        while (true) {
            try {
                System.out.print("Pilih Aksesori: ");
                pilihan = scanner.nextInt();
                if (pilihan < 1 || pilihan > dataAksesori.size()) {
                    System.out.println("Pilihan tidak valid. Silakan pilih aksesori yang tersedia.");
                    continue;
                }
                scanner.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Input tidak valid. Harap masukkan angka untuk memilih aksesori.");
                scanner.nextLine();
            }
        }
```

CRUD.java

```
        Aksesori aksesori = dataAksesori.get(pilihan - 1);

        System.out.print("Nama: ");
        String nama = scanner.nextLine();
        System.out.print("Merek: ");
        String merek = scanner.nextLine();

        int harga = 0;
        while (true) {
            try {
                System.out.print("Harga: ");
                harga = scanner.nextInt();
                scanner.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Input tidak valid. Harap masukkan angka untuk harga.");
                scanner.nextLine();
            }
        }

        System.out.println("Pilih Kategori: ");
        lihat_kategori();

        int pilihan_kategori = 0;
        while (true) {
            try {
                System.out.print("Pilihan: ");
                pilihan_kategori = scanner.nextInt();
                if (pilihan_kategori < 1 || pilihan_kategori > dataKategori.size()) {
                    System.out.println("Pilihan tidak valid. Silakan pilih kategori yang tersedia.");
                    continue;
                }
                scanner.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Input tidak valid. Harap masukkan angka untuk pilihan kategori.");
                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());
            }
        }
    }
}
```

Main (Static & Exception Handling)

Main.java

```java
import java.util.ArrayList;
import java.util.Scanner;
import model.Admin;
import model.Pengguna;
import service.CRUD;

public class Main {
    public static final ArrayList<Pengguna> dataPengguna = new ArrayList<>();  5 usages
    public static final ArrayList<Admin> dataAdmin = new ArrayList<>();  3 usages

    private static final Scanner scanner = new Scanner(System.in);  16 usages

    public static void main(String[] args) {
        Admin admin = new Admin( email: "Admin@mail.com",  username: "admin",  password: "admin");
        dataAdmin.add(admin);
        dataPengguna.add(Pengguna.DEFAULT_USER);

        while (true) {
            try {
                System.out.println("\n=== Menu Utama ===");
                System.out.println("1. Login");
                System.out.println("2. Register");
                System.out.println("3. Exit");
                System.out.print("Pilihan: ");
                int pilihan = scanner.nextInt();
                scanner.nextLine();

                switch (pilihan) {
                    case 1 -> login();
                    case 2 -> register();
                    case 3 -> {
                        System.out.println("Terima kasih telah menggunakan program ini.");
                        return;
                    }
                    default -> System.out.println("Pilihan tidak valid.");
                }
            } catch (Exception e) {
                System.out.println("Input harus berupa angka. ");
                scanner.nextLine();
            }
        }
    }

    private static void login() {  1 usage
        System.out.println("\n=== Login ===");
        System.out.print("Username: ");
        String username = scanner.nextLine();
        System.out.print("Password: ");
        String password = scanner.nextLine();

        for (Admin admin : dataAdmin) {
            if (username.equals(admin.getUsername()) && password.equals(admin.getPassword())) {
                System.out.println("\nLogin sebagai Admin berhasil!");
                menuAdmin();
                return;
            }
        }

        for (Pengguna pengguna : dataPengguna) {
            if (username.equals(pengguna.getUsername()) && password.equals(pengguna.getPassword())) {
                System.out.println("\nLogin sebagai User berhasil!");
                menuUser();
                return;
            }
        }

        System.out.println("Username atau password salah.");
    }

    private static void register() {  1 usage
        System.out.println("\n=== Register ===");
        System.out.print("Email: ");
        String email = scanner.nextLine();
        System.out.print("Username: ");
        String username = scanner.nextLine();
        System.out.print("Password: ");
        String password = scanner.nextLine();
        System.out.print("Masukkan Kode Registrasi Admin (Isi 0 untuk abaikan): ");
        int kodeAdmin = scanner.nextInt();
        scanner.nextLine();

        register(email, username, password, kodeAdmin);
    }
```

Main.java

```java
    private static void register(String email, String username, String password, int kodeAdmin) {
        if (kodeAdmin == 321123) {
            Admin admin = new Admin(email, username, password);
            dataAdmin.add(admin);
            System.out.println("Registrasi Admin berhasil!");
        } else if (kodeAdmin == 0) {
            Pengguna pengguna = new Pengguna(email, username, password);
            dataPengguna.add(pengguna);
            System.out.println("Registrasi Pengguna berhasil!");
        } else {
            System.out.println("Kode rahasia tidak valid. Registrasi gagal.");
        }
    }

    private static void menuAdmin() {  1 usage
        int pilihan;
        do {
            try {
                System.out.println("\n=== Menu Admin ===");
                System.out.println("1. Tambah Aksesori");
                System.out.println("2. Lihat Aksesori");
                System.out.println("3. Ubah Aksesori");
                System.out.println("4. Hapus Aksesori");
                System.out.println("5. Tambah Kategori");
                System.out.println("6. Lihat Kategori");
                System.out.println("7. Daftar Pengguna");
                System.out.println("8. Keluar");
                System.out.print("Pilihan: ");
                pilihan = scanner.nextInt();
                scanner.nextLine();

                switch (pilihan) {
                    case 1 -> CRUD.tambah_aksesori();
                    case 2 -> CRUD.lihat_aksesori();
                    case 3 -> CRUD.ubah_aksesori();
                    case 4 -> CRUD.hapus_aksesori();
                    case 5 -> CRUD.tambah_kategori();
                    case 6 -> CRUD.lihat_kategori();
                    case 7 -> {
                        System.out.println("\n=== Daftar Pengguna ===");
                        if (dataPengguna.isEmpty()) {
                            System.out.println("Tidak ada pengguna yang terdaftar.");
                        } else {
                            for (Pengguna pengguna : dataPengguna) {
                                System.out.println(pengguna.toString());
                            }
                        }
                    }
                    case 8 -> {
                        System.out.println("Keluar dari menu admin.");
                        return;
                    }
                    default -> System.out.println("Pilihan tidak valid.");
                }
            } catch (Exception e) {
                System.out.println("Input harus berupa angka. ");
                scanner.nextLine();
            }
        } while (true);
    }

    private static void menuUser() {  1 usage
        try {
            System.out.println("\n=== Menu User ===");
            System.out.println("1. Lihat Aksesori");
            System.out.println("2. Keluar");
            System.out.print("Pilihan: ");
            int pilihan = scanner.nextInt();
            scanner.nextLine();

            if (pilihan == 1) {
                CRUD.lihat_aksesori();
            } else if (pilihan == 2) {
                System.out.println("Keluar dari menu user.");
            } else {
                System.out.println("Pilihan tidak valid.");
            }
        } catch (Exception e) {
            System.out.println("Input harus berupa angka. ");
            scanner.nextLine();
        }
    }
}
```