

Nama :Ahmadyani

Nim :2309106055

POSTTEST 6

A. SOURCECODE

```
import java.util.ArrayList;
import java.util.InputMismatchException;
import java.util.Scanner;

// Interface
interface Klinik {
    void registrasi();
    void konsultasi();
}

// Abstract class
abstract class User {
    protected final String id;
    protected String nama;

    public User(String id, String nama) {
        this.id = id;
        this.nama = nama;
    }

    public abstract void tampilkanData();

    public final void tampilkanData(String prefix) {
        System.out.println(prefix + "ID: " + id + ", Nama: " + nama);
    }
}

// Pasien mengimplementasikan interface Klinik
class Pasien extends User implements Klinik {
    private int usia;
    private static int totalPasien = 0;

    public Pasien(String idPasien, String nama, int usia) {
        super(idPasien, nama);
        this.usia = usia;
    }
}
```

```

        totalPasien++;
    }

    public int getUsia() {
        return usia;
    }

    public void setUsia(int usia) {
        if (usia > 0) {
            this.usia = usia;
        }
    }

    @Override
    public void tampilkanData() {
        System.out.println("Pasien - ID: " + id + ", Nama: " + nama + ", Usia: " + usia);
    }

    @Override
    public void registrasi() {
        System.out.println(nama + " berhasil registrasi di klinik.");
    }

    @Override
    public void konsultasi() {
        System.out.println(nama + " sedang melakukan konsultasi.");
    }

    public static int getTotalPasien() {
        return totalPasien;
    }
}

class Dokter extends User {
    private String spesialisasi;
    private static int totalDokter = 0;

    public Dokter(String idDokter, String nama, String spesialisasi) {
        super(idDokter, nama);
        this.spesialisasi = spesialisasi;
        totalDokter++;
    }

    public String getSpesialisasi() {
        return spesialisasi;
    }
}

```

```

    public void setSpesialisasi(String spesialisasi) {
        this.spesialisasi = spesialisasi;
    }

    @Override
    public void tampilkanData() {
        System.out.println("Dokter - ID: " + id + ", Nama: " + nama + ",
Spesialisasi: " + spesialisasi);
    }

    public static int getTotalDokter() {
        return totalDokter;
    }
}

public final class Ahmadyani_2309106055_Posttest5 {
    static ArrayList<Pasien> daftarPasien = new ArrayList<>();
    static ArrayList<Dokter> daftarDokter = new ArrayList<>();
    static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        int pilihan;
        do {
            System.out.println("\nMenu:");
            System.out.println("1. Tambah Pasien");
            System.out.println("2. Tampilkan Pasien");
            System.out.println("3. Hapus Pasien");
            System.out.println("4. Tambah Dokter");
            System.out.println("5. Tampilkan Dokter");
            System.out.println("6. Hapus Dokter");
            System.out.println("7. Coba Polymorphism");
            System.out.println("8. Total Pasien dan Dokter");
            System.out.println("9. Keluar");
            System.out.print("Pilih menu: ");
            try {
                pilihan = scanner.nextInt();
                scanner.nextLine();

                switch (pilihan) {
                    case 1 -> tambahPasien();
                    case 2 -> tampilkanPasien();
                    case 3 -> hapusPasien();
                    case 4 -> tambahDokter();
                    case 5 -> tampilkanDokter();
                    case 6 -> hapusDokter();
                    case 7 -> cobaPolymorphism();
                    case 8 -> tampilkanTotal();
                    case 9 -> System.out.println("Keluar...");
                }
            } catch (Exception e) {
                System.out.println("Invalid input. Please enter a number between 1 and 9.");
            }
        } while (pilihan != 9);
    }
}

```

```

        default -> System.out.println("Pilihan tidak valid.");
    }
} catch (InputMismatchException e) {
    System.out.println("Input harus berupa angka!");
    scanner.nextLine(); // membersihkan buffer
    pilihan = 0;
}
} while (pilihan != 9);
}

static void tambahPasien() {
    try {
        System.out.print("ID Pasien: ");
        String id = scanner.nextLine();
        System.out.print("Nama: ");
        String nama = scanner.nextLine();
        System.out.print("Usia: ");
        int usia = scanner.nextInt(); scanner.nextLine();
        Pasien p = new Pasien(id, nama, usia);
        daftarPasien.add(p);
        System.out.println("Pasien ditambahkan.");
        p.registrasi();
    } catch (InputMismatchException e) {
        System.out.println("Usia harus berupa angka.");
        scanner.nextLine();
    }
}

static void tampilkanPasien() {
    if (daftarPasien.isEmpty()) {
        System.out.println("Tidak ada data pasien.");
    } else {
        for (Pasien p : daftarPasien) {
            p.tampilkanData();
        }
    }
}

static void hapusPasien() {
    System.out.print("Masukkan ID Pasien yang ingin dihapus: ");
    String id = scanner.nextLine();
    boolean ditemukan = false;
    for (int i = 0; i < daftarPasien.size(); i++) {
        if (daftarPasien.get(i).id.equals(id)) {
            daftarPasien.remove(i);
            System.out.println("Pasien dengan ID " + id + " telah
dihapus.");
            ditemukan = true;

```

```

        break;
    }
}
if (!ditemukan) {
    System.out.println("Pasien dengan ID " + id + " tidak
ditemukan.");
}
}

static void tambahDokter() {
    System.out.print("ID Dokter: ");
    String id = scanner.nextLine();
    System.out.print("Nama: ");
    String nama = scanner.nextLine();
    System.out.print("Spesialisasi: ");
    String spesialisasi = scanner.nextLine();
    daftarDokter.add(new Dokter(id, nama, spesialisasi));
    System.out.println("Dokter ditambahkan.");
}

static void tampilkanDokter() {
    if (daftarDokter.isEmpty()) {
        System.out.println("Tidak ada data dokter.");
    } else {
        for (Dokter d : daftarDokter) {
            d.tampilkanData();
        }
    }
}

static void hapusDokter() {
    System.out.print("Masukkan ID Dokter yang ingin dihapus: ");
    String id = scanner.nextLine();
    boolean ditemukan = false;
    for (int i = 0; i < daftarDokter.size(); i++) {
        if (daftarDokter.get(i).id.equals(id)) {
            daftarDokter.remove(i);
            System.out.println("Dokter dengan ID " + id + " telah
dihapus.");
            ditemukan = true;
            break;
        }
    }
    if (!ditemukan) {
        System.out.println("Dokter dengan ID " + id + " tidak
ditemukan.");
    }
}
}

```

```

static void
cobaPolymorphism() {
    User u1 = new Pasien("P01", "Yani", 21);
    User u2 = new Dokter("D01", "Dr. Budi", "Anak");
    u1.tampilkanData();
    u2.tampilkanData();
    u1.tampilkanData(">>> ");
    ((Pasien) u1).konsultasi();
}

static void tampilkanTotal() {
    System.out.println("Total Pasien: " + Pasien.getTotalPasien());
    System.out.println("Total Dokter: " + Dokter.getTotalDokter());
}
}

```

B. SCREENSHOT DEMO PROGRAM

Menu:
1. Tambah Pasien
2. Tampilkan Pasien
3. Hapus Pasien
4. Tambah Dokter
5. Tampilkan Dokter
6. Hapus Dokter
7. Coba Polymorphism
8. Total Pasien dan Dokter
9. Keluar
Pilih menu:

Pilih menu: 1
ID Pasien: 3
Nama: amay
Usia: 88
Pasien ditambahkan.
amay berhasil registrasi di klinik.

Pilih menu: 2
Pasien - ID: 3, Nama: amay, Usia: 88

Pilih menu: 3
Masukkan ID Pasien yang ingin dihapus: 3
Pasien dengan ID 3 telah dihapus.

Pilih menu: 4
ID Dokter: 3
Nama: yaya
Spesialisasi: gigi
Dokter ditambahkan.

Menu:
Pilih menu: 6
Masukkan ID Dokter yang ingin dihapus: 3
Dokter dengan ID 3 telah dihapus.

```
Pilih menu: 7
Pasien - ID: P01, Nama: Yani, Usia: 21
Dokter - ID: D01, Nama: Dr. Budi, Spesialisasi: Anak
>>> ID: P01, Nama: Yani
Yani sedang melakukan konsultasi.
```

```
Pilih menu: 8
Total Pasien: 2
Total Dokter: 2
```

```
Pilih menu: 9
Keluar...
```