

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK  
POSTTEST 4**



**Informatika A'23  
Nur Juzieatul Alifah  
2309106040**

**PROGRAM STUDI INFORMATIKA  
UNIVERSITAS MULAWARMAN  
SAMARINDA  
2024**

```
package model;

public class Dokter extends Orang implements TindakanMedis {
    private final int id;
    private String spesialis;

    public static int jumlahDokter = 0;

    public Dokter(int id, String nama, String spesialis, String noTelp, String email) {
        super(nama, noTelp, email);
        this.id = id;
        this.spesialis = spesialis;
        jumlahDokter++;
    }

    public int getId() { return id; }
    public String getSpesialis() { return spesialis; }
    public void setSpesialis(String spesialis) { this.spesialis = spesialis; }

    public static void tampilkanJumlahDokter() {
        System.out.println("Jumlah dokter terdaftar: " + jumlahDokter);
    }

    @Override
    public String toString() {
        return "ID Dokter: " + id + ", " + super.nama + ", Spesialis: " + spesialis;
    }

    @Override
    public void tampilkanProfil() {
        System.out.println("Profil Dokter: " + toString());
    }

    @Override
    public void beriDiagnosis(String diagnosis) {
        if (diagnosis == null || diagnosis.isBlank()) {
            System.err.println("Error: Diagnosis tidak boleh kosong.");
            return;
        }
        System.out.println("Dokter " + nama + " memberikan diagnosis: " + diagnosis);
    }

    @Override
    public void beriResep(String obat) {
        if (obat == null || obat.isBlank()) {
            System.err.println("Error: Obat tidak boleh kosong.");
            return;
        }
        System.out.println("Dokter " + nama + " meresepkan obat: " + obat);
    }
}
```

```
package model;

public interface TindakanMedis {
    void beriDiagnosis(String diagnosis);
    void beriResep(String obat);
}
```

```

public class Main {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Dokter2 dokterManager = new Dokter2();
        Jadwal2 jadwalManager = new Jadwal2();
        RekamMedis2 rekamMedisManager = new RekamMedis2();
        ArrayList<Pasien> daftarPasien = new ArrayList<>();

        while (true) {
            System.out.println(x:"\n===== Menu =====");
            System.out.println(x:"1. Tambah Dokter");
            System.out.println(x:"2. Tampilkan Dokter");
            System.out.println(x:"3. Edit Dokter");
            System.out.println(x:"4. Hapus Dokter");
            System.out.println(x:"5. Tambah Jadwal");
            System.out.println(x:"6. Tampilkan Jadwal");
            System.out.println(x:"7. Edit Jadwal");
            System.out.println(x:"8. Hapus Jadwal");
            System.out.println(x:"9. Tambah Rekam Medis");
            System.out.println(x:"10. Tampilkan Rekam Medis");
            System.out.println(x:"11. Edit Rekam Medis");
            System.out.println(x:"12. Hapus Rekam Medis");
            System.out.println(x:"13. Polymorphism (Pasien)");
            System.out.println(x:"14. Tambah Pasien");
            System.out.println(x:"15. Tampilkan Pasien");
            System.out.println(x:"16. Edit Pasien");
            System.out.println(x:"17. Hapus Pasien");
            System.out.println(x:"18. Diagnosa dan Resep Obat");
            System.out.println(x:"19. Keluar");
            System.out.print(s:"Pilih menu: ");

            int pilihan;
            try {
                pilihan = Integer.parseInt(scanner.nextLine());
            } catch (NumberFormatException e) {
                System.out.println(x:"Input harus berupa angka. Silakan coba lagi.");
                continue;
            }

            switch (pilihan) {
                case 1 -> dokterManager.tambahDokter(scanner);
                case 2 -> dokterManager.tampilkanDokter();
                case 3 -> dokterManager.editDokter(scanner);
                case 4 -> dokterManager.hapusDokter(scanner);
                case 5 -> jadwalManager.tambahJadwal(scanner);
                case 6 -> jadwalManager.tampilkanJadwal();
                case 7 -> jadwalManager.editJadwal(scanner);
                case 8 -> jadwalManager.hapusJadwal(scanner);
                case 9 -> rekamMedisManager.tambahRekamMedis(scanner);
                case 10 -> rekamMedisManager.tampilkanRekamMedis();
                case 11 -> rekamMedisManager.editRekamMedis(scanner);
                case 12 -> rekamMedisManager.hapusRekamMedis(scanner);
                case 13 -> {
                    Pasien contohPasien = new Pasien(idPasien:101, nama:"Elly", usia:22, alamat:"Jl. Kenanga", noTelp:"0822334455", email:"elly@mail.com");
                    System.out.println(x:"\nTampilan info lengkap:");
                    contohPasien.tampilkanInfo();
                    System.out.println(x:"\nTampilan info singkat:");
                    contohPasien.tampilkanInfo(singkat:true);
                }
            }
        }
    }
}

```

```

switch (pilihan) {
    case 1 -> dokterManager.tambahDokter(scanner);
    case 2 -> dokterManager.tampilkanDokter();
    case 3 -> dokterManager.editDokter(scanner);
    case 4 -> dokterManager.hapusDokter(scanner);
    case 5 -> jadwalManager.tambahJadwal(scanner);
    case 6 -> jadwalManager.tampilkanJadwal();
    case 7 -> jadwalManager.editJadwal(scanner);
    case 8 -> jadwalManager.hapusJadwal(scanner);
    case 9 -> rekamMedisManager.tambahRekamMedis(scanner);
    case 10 -> rekamMedisManager.tampilkanRekamMedis();
    case 11 -> rekamMedisManager.editRekamMedis(scanner);
    case 12 -> rekamMedisManager.hapusRekamMedis(scanner);
    case 13 -> {
        Pasien contohPasien = new Pasien(idPasien:101, nama:"Elly", usia:22, alamat:"Jl. Kenanga", noTelp:"0822334455", email:"elly@mail.com");
        System.out.println(x:\nTampilan info lengkap:");
        contohPasien.tampilkanInfo();
        System.out.println(x:\nTampilan info singkat:");
        contohPasien.tampilkanInfo(singkat:true);
    }
}

case 15 -> {
    if (daftarPasien.isEmpty()) {
        System.out.println(x:Belum ada data pasien.");
    } else {
        for (Pasien p : daftarPasien) {
            p.tampilkanInfo();
            System.out.println(x:-----");
        }
    }
}
case 16 -> {
    try {
        System.out.print(s:"Masukkan ID pasien yang ingin diedit: ");
        int editId = Integer.parseInt(scanner.nextLine());
        boolean ditemukan = false;

        for (Pasien p : daftarPasien) {
            if (p.getIdPasien() == editId) {
                System.out.print(s:"Nama baru: ");
                p.setNama(scanner.nextLine());
                System.out.print(s:"Usia baru: ");
                p.setUsia(Integer.parseInt(scanner.nextLine()));
                System.out.print(s:"Alamat baru: ");
                p.setAlamat(scanner.nextLine());
                System.out.print(s:"No Telp baru: ");
                p.setNoTelp(scanner.nextLine());
                System.out.print(s:"Email baru: ");
                p.setEmail(scanner.nextLine());
                System.out.println(x:Data pasien berhasil diperbarui.");
                ditemukan = true;
                break;
            }
        }
        if (!ditemukan) {
            System.out.println(x:Pasien dengan ID tersebut tidak ditemukan.");
        }
    } catch (NumberFormatException e) {
        System.out.println(x:Input angka tidak valid.");
    }
}

case 17 -> {
    try {
        System.out.print(s:"Masukkan ID pasien yang ingin dihapus: ");
        int hapusId = Integer.parseInt(scanner.nextLine());
        boolean terhapus = daftarPasien.removeIf(p -> p.getIdPasien() == hapusId);
        if (terhapus) {
            System.out.println(x:Pasien berhasil dihapus.");
        } else {
            System.out.println(x:Pasien dengan ID tersebut tidak ditemukan.");
        }
    } catch (NumberFormatException e) {
        System.out.println(x:Input ID tidak valid.");
    }
}
case 18 -> {
    Dokter d1 = new Dokter(id:1, nama:"Dr. Andi", spesialis:"Saraf", noTelp:"0812345678", email:"andi@klinik.com");
    d1.tampilkanProfil();
    d1.beriDiagnosis(diagnosis:"Migrain");
    d1.beriResep(obat:"Paracetamol");
    Dokter.tampilkanJumlahDokter();
}
case 19 -> {
    System.out.println(x:Bye bye");
    scanner.close();
    return;
}
default -> System.out.println(x:Pilihan tidak valid. Silakan coba lagi.");
}

```