



UJIAN TENGAH SEMESTER (UTS)
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)

PETUNJUK Pengerjaan

1. Nama project **UTS_SIRAMPASIEN** ditambah nomor urut presensi (contoh : **UTS_SIRAMPASIEN01**)
2. Semua nama class ditambah nomor urut presensi (contoh : **Dokter01, Pasien01**)
3. Semua nama variabel/attribute ditambah nomor urut presensi (contoh : **id01, nama01**)
4. File yang harus dikumpulkan dalam bentuk kompresi rar / zip dengan format penamaan **UTS_NomorUrutPresensi_NamaMahasiswa**
5. Lampirkan screenshot script dan hasil running program secara lengkap dengan minimal 5 data Dokter dan Pasien, serta minimal 2 data Riwayat Pemeriksaan atau Rekam Medik setiap Pasien

SOAL

Buatlah **Sistem Informasi Rekam Medik Pasien (SIRAM PASIEN)** di sebuah klinik kesehatan dengan menerapkan konsep Pemrograman Berorientasi Objek (PBO). Sistem informasi ini akan mencatat beberapa data Dokter, Pasien dan Riwayat Pemeriksaan atau Rekam Mediknya. Pada sistem informasi ini melibatkan beberapa class sebagai berikut;

1. **Orang** dengan attribute dan method sebagai berikut;

```
public class Orang {  
    private String nama;  
    private boolean jenisKelamin;  
    private String alamat;  
    private String noTelp;  
    public Orang(String nama, Boolean jenisKelamin, String alamat,  
        String noTelp) {  
        this.nama = nama;  
        this.jenisKelamin = jenisKelamin;  
        this.alamat = alamat;  
        this.noTelp = noTelp;  
    }  
    // Terapkan Teknik Overloading jika dimungkinkan  
    // Setter methods  
    .....  
    // Getter methods  
    .....  
}
```



2. **Pasien** dengan attribute dan method sebagai berikut;

```
public class Pasien extends Orang{
    private int id;
    private RekamMedik rekamMedik;

    //Terapkan teknik Overloading dan Overriding jika dimungkinkan
    //Gunakan keyword super jika dimungkinkan
    public Pasien(int id, String nama, Boolean jenisKelamin, String
        alamat, String noTelp) {
        .....
        .....
        .....
        .....
    }
    // Setter methods
    public void setRekamMedik(RekamMedik rekamMedik) {
        this.rekamMedik = rekamMedik;
    }
    // Getter methods
    .....
}
```

3. **Dokter** dengan attribute dan method sebagai berikut;

```
public class Dokter extends Orang {
    private int id;
    private String spesialisasi;

    //Terapkan teknik Overloading dan Overriding jika dimungkinkan
    //Gunakan keyword super jika dimungkinkan
    public Dokter(int id, String nama, Boolean jenisKelamin, String
        alamat, String noTelp, String spesialisasi) {
        .....
        .....
        .....
        .....
    }
    // Getter methods
    .....
}
```

4. **Pemeriksaan** dengan attribute dan method sebagai berikut;

```
import java.util.Date;
public class Pemeriksaan {
    private int id;
    private Date tanggalPemeriksaan;
    private String keluhan;
    private String diagnosa;
    private String tindakan;
    private String obat;
    private Dokter dokter;
```



```
public Pemeriksaan(int id, Date tanggalPemeriksaan, String keluhan,
    String diagnosa, String tindakan, String obat, Dokter dokter) {
    this.id = id;
    this.tanggalPemeriksaan = tanggalPemeriksaan;
    this.keluhan = keluhan;
    this.diagnosa = diagnosa;
    this.tindakan = tindakan;
    this.obat = obat;
    this.dokter = dokter;
}
// Getter methods
.....
}
```

5. **RekamMedik** dengan attribute dan method sebagai berikut;

```
import java.util.ArrayList;
import java.util.List;
public class RekamMedik {
    private int id;
    private Pasien pasien;
    private List<Pemeriksaan> daftarPemeriksaan;
    public RekamMedik(int id, Pasien pasien) {
        this.id = id;
        this.pasien = pasien;
        this.daftarPemeriksaan = new ArrayList<>();
    }
    public void tambahPemeriksaan(Pemeriksaan pemeriksaan) {
        if (daftarPemeriksaan == null) {
            daftarPemeriksaan = new ArrayList<>();
        }
        daftarPemeriksaan.add(pemeriksaan);
    }
    public void setDaftarPemeriksaan(List<Pemeriksaan>
        daftarPemeriksaan) {
        this.daftarPemeriksaan = daftarPemeriksaan;
    }
    public List<Pemeriksaan> getDaftarPemeriksaan() {
        return daftarPemeriksaan; // Mengembalikan referensi ke list
    }
}
```



6. **SIRAM_PASIEN** sebagai class utama (*main class*) dengan ketentuan sebagai berikut;

6.1. Menampilkan menu program sebagai berikut;

Menu Program;

1. **Tambah Pasien**
 2. **Update Data Pasien**
 3. **Tampilkan Daftar Pasien**
 4. **Tambah Dokter**
 5. **Update Data Dokter**
 6. **Tampilkan Daftar Dokter**
 7. **Tambah Riwayat Pemeriksaan (Rekam Medik) Pasien**
 8. **Tampilkan Riwayat Pemeriksaan (Rekam Medik) Pasien**
 9. **Selesai**
- Pilih Menu Nomer ?

6.2. Setiap pilihan menu akan menjalankan sebuah method tertentu yang sesuai, misalkan;

- Jika dipilih menu 1 (**Tambah Pasien**), maka menjalankan method **tambahPasien()**
- Jika dipilih menu 5 (**Tampilkan Daftar Dokter**), maka menjalankan method **tampilkanDaftarDokter()**
- Jika dipilih menu 9 (**Selesai**), maka program berakhir

6.3. Semua data dimasukkan (di-input) melalui keyboard, seperti contoh berikut;

```
Scanner scanner = new Scanner(System.in);
DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("dd/MM/yyyy");
System.out.print("Masukkan tanggal (dd/MM/yyyy): ");
String inputTanggal = scanner.nextLine();
```

6.4. Terapkan validasi inputan untuk setiap nilai atribut yang akan disimpan sesuai dengan tipe datanya (int, String atau Date). Mekanisme validasi inputan data dapat menggunakan metode **try catch**

6.5. Formulir isian data **Dokter, Pasien dan Pemeriksaan (Rekam Medik)** disesuaikan dengan attribute yang terdapat pada masing-masing class.

6.6. Untuk method setter maupun getter pada masing-masing class silahkan dikembangkan sesuai kebutuhan.

6.7. Gunakan Teknik **Overriding** dan **Overloading** serta keyword **super** pada masing-masing class (dikembangkan sesuai kebutuhan).