|  |  |  |
| --- | --- | --- |
| **Nama** | : | Vincentius Leonanda Prbowo |
| **Nim** | : | 2341720149 |
| **Kelas** | : | TI-2A |

**LEMBAR JAWABAN UTS**

|  |  |
| --- | --- |
| **Soal No** | **Jawaban** |
| 1 | Penggunaan Overloading |
| 2 | Penggunaan Overriding |
| 3 | Penggunaan Try Catch |
| 4 | Output menu : |
| 5 | Output Tambah Pasien : |
| 6 | Output Update Data Pasien : |
| 7 | Output Tampilkan Daftar Pasien : |
| 8 | Output Tambah Dokter : |
| 9 | Output Update Data Dokter : |
| 10 | Output Tampilkan Data Dokter : |
| 11 | Output Tambah Riwayat Pasien : |
| 12 | Output Tampilkan Riwayat Pasien : |
| 13 | Output Keluar : |
| 14 | Code Class Dokter28 :  public class Dokter28 extends Orang28 {      private int id;      private String spesialisasi;      public Dokter28(int id, String nama, Boolean jenisKelamin, String alamat, String noTelp, String spesialisasi) {          super(nama, jenisKelamin, alamat, noTelp);          this.id = id;          this.spesialisasi = spesialisasi;      }      // Overriding method getNama() dari class Orang28      @Override      public String getNama() {          return "Dr. " + super.getNama();      }      public int getId() {          return id;      }      public String getSpesialisasi() {          return spesialisasi;      }  } |
| 15 | public class Orang28 {      private String nama;      private boolean jenisKelamin;      private String alamat;      private String noTelp;      public Orang28() {      }      public Orang28(String nama, Boolean jenisKelamin, String alamat, String noTelp) {          this.nama = nama;          this.jenisKelamin = jenisKelamin;          this.alamat = alamat;          this.noTelp = noTelp;      }      public Orang28(String nama, Boolean jenisKelamin, String alamat) {          this.nama = nama;          this.jenisKelamin = jenisKelamin;          this.alamat = alamat;      }      public Orang28(String nama, Boolean jenisKelamin) {          this.nama = nama;          this.jenisKelamin = jenisKelamin;      }      public String getAlamat() {          return alamat;      }      public String getNama() {          return nama;      }      public String getNoTelp() {          return noTelp;      }  } |
| 16 | public class Pasien28 extends Orang28 {      private int id;      private RekamMedik28 rekamMedik;      public Pasien28(int id, String nama, Boolean jenisKelamin, String alamat, String noTelp) {          super(nama, jenisKelamin, alamat, noTelp);          this.id = id;      }      // Overriding method getNama() dari class Orang28      @Override      public String getNama() {          return "Pasien: " + super.getNama();      }      public int getId() {          return id;      }      public RekamMedik28 getRekamMedik() {          return rekamMedik;      }      public void setRekamMedik(RekamMedik28 rekamMedik) {          this.rekamMedik = rekamMedik;      }  } |
| 17 | import java.util.Date;  public class Pemeriksaan28 {      private int id;      private Date tanggalPemeriksaan;      private String keluhan;      private String diagnosa;      private String tindakan;      private String obat;      private Dokter28 dokter;      public Pemeriksaan28(int id, Date tanggalPemeriksaan, String keluhan,              String diagnosa, String tindakan, String obat, Dokter28 dokter) {          this.id = id;          this.tanggalPemeriksaan = tanggalPemeriksaan;          this.keluhan = keluhan;          this.diagnosa = diagnosa;          this.tindakan = tindakan;          this.obat = obat;          this.dokter = dokter;      }      public int getId() {          return id;      }      public Date getTanggalPemeriksaan() {          return tanggalPemeriksaan;      }      public String getKeluhan() {          return keluhan;      }      public String getDiagnosa() {          return diagnosa;      }      public String getTindakan() {          return tindakan;      }      public String getObat() {          return obat;      }      public Dokter28 getDokter() {          return dokter;      }  } |
| 18 | import java.util.ArrayList;  import java.util.List;  public class RekamMedik28 {      private int id;      private Pasien28 pasien;      private List<Pemeriksaan28> daftarPemeriksaan;      public RekamMedik28(int id, Pasien28 pasien) {          this.id = id;          this.pasien = pasien;          this.daftarPemeriksaan = new ArrayList<>();      }      public void tambahPemeriksaan(Pemeriksaan28 pemeriksaan) {          if (daftarPemeriksaan == null) {              daftarPemeriksaan = new ArrayList<>();          }          daftarPemeriksaan.add(pemeriksaan);      }      public void setDaftarPemeriksaan(List<Pemeriksaan28> daftarPemeriksaan) {          this.daftarPemeriksaan = daftarPemeriksaan;      }      public List<Pemeriksaan28> getDaftarPemeriksaan() {          return daftarPemeriksaan; // Mengembalikan referensi ke list      }  } |
| 19 | import java.util.Scanner;  import java.util.ArrayList;  import java.util.List;  import java.time.format.DateTimeFormatter;  import java.time.LocalDate;  public class SIRAM\_PASIEN {      private static List<Pasien28> daftarPasien = new ArrayList<>();      private static List<Dokter28> daftarDokter = new ArrayList<>();      public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          int pilihan;          do {              System.out.println("Menu Program:");              System.out.println("1. Tambah Pasien");              System.out.println("2. Update Data Pasien");              System.out.println("3. Tampilkan Daftar Pasien");              System.out.println("4. Tambah Dokter");              System.out.println("5. Update Data Dokter");              System.out.println("6. Tampilkan Daftar Dokter");              System.out.println("7. Tambah Riwayat Pemeriksaan (Rekam Medik) Pasien");              System.out.println("8. Tampilkan Riwayat Pemeriksaan (Rekam Medik) Pasien");              System.out.println("9. Selesai");              System.out.print("Pilih Menu Nomer ? …… ");              try {                  pilihan = scanner.nextInt();                  scanner.nextLine(); // Mengambil sisa new line                  switch (pilihan) {                      case 1:                          tambahPasien(scanner);                          break;                      case 2:                          updatePasien(scanner);                          break;                      case 3:                          tampilkanDaftarPasien();                          break;                      case 4:                          tambahDokter(scanner);                          break;                      case 5:                          updateDokter(scanner);                          break;                      case 6:                          tampilkanDaftarDokter();                          break;                      case 7:                          tambahPemeriksaan(scanner);                          break;                      case 8:                          tampilkanRiwayatPemeriksaan(scanner);                          break;                      case 9:                          System.out.println("Program selesai.");                          break;                      default:                          System.out.println("Pilihan tidak valid. Coba lagi.");                  }              } catch (Exception e) {                  System.out.println("Input tidak valid, silakan coba lagi.");                  scanner.nextLine(); // Bersihkan input yang salah              }          } while (true);      }      // Method tambahPasien      public static void tambahPasien(Scanner scanner) {          try {              System.out.print("Masukkan ID Pasien: ");              int id = Integer.parseInt(scanner.nextLine()); // Validasi input angka              System.out.print("Masukkan Nama Pasien: ");              String nama = scanner.nextLine();              System.out.print("Masukkan Jenis Kelamin (true untuk laki-laki, false untuk perempuan): ");              boolean jenisKelamin = Boolean.parseBoolean(scanner.nextLine());              System.out.print("Masukkan Alamat: ");              String alamat = scanner.nextLine();              System.out.print("Masukkan Nomor Telepon: ");              String noTelp = scanner.nextLine();              Pasien28 pasien = new Pasien28(id, nama, jenisKelamin, alamat, noTelp);              daftarPasien.add(pasien);              System.out.println("Pasien berhasil ditambahkan.");          } catch (NumberFormatException e) {              System.out.println("Input ID tidak valid! Harus berupa angka.");          } catch (Exception e) {              System.out.println("Terjadi kesalahan: " + e.getMessage());          }      }      // Method tambahDokter      public static void tambahDokter(Scanner scanner) {          try {              System.out.print("Masukkan ID Dokter: ");              int id = Integer.parseInt(scanner.nextLine()); // Validasi input angka              System.out.print("Masukkan Nama Dokter: ");              String nama = scanner.nextLine();              System.out.print("Masukkan Jenis Kelamin (true untuk laki-laki, false untuk perempuan): ");              boolean jenisKelamin = Boolean.parseBoolean(scanner.nextLine());              System.out.print("Masukkan Alamat: ");              String alamat = scanner.nextLine();              System.out.print("Masukkan Nomor Telepon: ");              String noTelp = scanner.nextLine();              System.out.print("Masukkan Spesialisasi: ");              String spesialisasi = scanner.nextLine();              Dokter28 dokter = new Dokter28(id, nama, jenisKelamin, alamat, noTelp, spesialisasi);              daftarDokter.add(dokter);              System.out.println("Dokter berhasil ditambahkan.");          } catch (NumberFormatException e) {              System.out.println("Input ID tidak valid! Harus berupa angka.");          } catch (Exception e) {              System.out.println("Terjadi kesalahan: " + e.getMessage());          }      }      // Method tambahPemeriksaan      public static void tambahPemeriksaan(Scanner scanner) {          try {              System.out.print("Masukkan ID Pasien: ");              int idPasien = Integer.parseInt(scanner.nextLine());              Pasien28 pasien = null;              for (Pasien28 p : daftarPasien) {                  if (p.getId() == idPasien) {                      pasien = p;                      break;                  }              }              if (pasien == null) {                  System.out.println("Pasien dengan ID " + idPasien + " tidak ditemukan.");                  return;              }              System.out.print("Masukkan ID Dokter yang memeriksa: ");              int idDokter = Integer.parseInt(scanner.nextLine());              Dokter28 dokter = null;              for (Dokter28 d : daftarDokter) {                  if (d.getId() == idDokter) {                      dokter = d;                      break;                  }              }              if (dokter == null) {                  System.out.println("Dokter dengan ID " + idDokter + " tidak ditemukan.");                  return;              }              System.out.print("Masukkan keluhan pasien: ");              String keluhan = scanner.nextLine();              System.out.print("Masukkan diagnosa: ");              String diagnosa = scanner.nextLine();              System.out.print("Masukkan tindakan: ");              String tindakan = scanner.nextLine();              System.out.print("Masukkan obat: ");              String obat = scanner.nextLine();              System.out.print("Masukkan tanggal pemeriksaan (dd/MM/yyyy): ");              String inputTanggal = scanner.nextLine();              DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd/MM/yyyy");              LocalDate tanggal = LocalDate.parse(inputTanggal, formatter);              Pemeriksaan28 pemeriksaan = new Pemeriksaan28(idPasien, java.sql.Date.valueOf(tanggal), keluhan, diagnosa,                      tindakan, obat, dokter);              RekamMedik28 rekamMedik = pasien.getRekamMedik();              if (rekamMedik == null) {                  rekamMedik = new RekamMedik28(idPasien, pasien);                  pasien.setRekamMedik(rekamMedik);              }              rekamMedik.tambahPemeriksaan(pemeriksaan);              System.out.println("Pemeriksaan berhasil ditambahkan untuk pasien ID: " + idPasien);          } catch (NumberFormatException e) {              System.out.println("Input ID atau Tanggal tidak valid! Harus berupa angka.");          } catch (Exception e) {              System.out.println("Terjadi kesalahan: " + e.getMessage());          }      }      public static void tampilkanDaftarPasien() {          if (daftarPasien.isEmpty()) {              System.out.println("Tidak ada pasien yang terdaftar.");          } else {              for (Pasien28 pasien : daftarPasien) {                  System.out.println("ID: " + pasien.getId() + ", Nama: " + pasien.getNama());              }          }      }      public static void tampilkanDaftarDokter() {          if (daftarDokter.isEmpty()) {              System.out.println("Tidak ada dokter yang terdaftar.");          } else {              for (Dokter28 dokter : daftarDokter) {                  System.out.println("ID: " + dokter.getId() + ", Nama: " + dokter.getNama());              }          }      }      public static void tampilkanRiwayatPemeriksaan(Scanner scanner) {          System.out.print("Masukkan ID Pasien: ");          int id = scanner.nextInt();          scanner.nextLine(); // ambil sisa new line          Pasien28 pasien = null;          for (Pasien28 p : daftarPasien) {              if (p.getId() == id) {                  pasien = p;                  break;              }          }          if (pasien == null) {              System.out.println("Pasien dengan ID " + id + " tidak ditemukan.");              return;          }          RekamMedik28 rekamMedik = pasien.getRekamMedik();          if (rekamMedik == null || rekamMedik.getDaftarPemeriksaan().isEmpty()) {              System.out.println("Tidak ada riwayat pemeriksaan untuk pasien ini.");              return;          }          System.out.println("Riwayat Pemeriksaan untuk Pasien ID: " + id);          for (Pemeriksaan28 pemeriksaan : rekamMedik.getDaftarPemeriksaan()) {              System.out.println("ID Pemeriksaan: " + pemeriksaan.getId());              System.out.println("Tanggal Pemeriksaan: " + pemeriksaan.getTanggalPemeriksaan());              System.out.println("Keluhan: " + pemeriksaan.getKeluhan());              System.out.println("Diagnosa: " + pemeriksaan.getDiagnosa());              System.out.println("Tindakan: " + pemeriksaan.getTindakan());              System.out.println("Obat: " + pemeriksaan.getObat());              System.out.println("Dokter: " + pemeriksaan.getDokter().getNama());              System.out.println("---------------------------");          }      }      public static void updateDokter(Scanner scanner) {          System.out.print("Masukkan ID Dokter yang ingin diupdate: ");          int id = scanner.nextInt();          scanner.nextLine(); // ambil sisa new line          for (Dokter28 dokter : daftarDokter) {              if (dokter.getId() == id) {                  System.out.print("Masukkan Nama Baru: ");                  String nama = scanner.nextLine();                  System.out.print("Masukkan Jenis Kelamin Baru (true untuk laki-laki, false untuk perempuan): ");                  boolean jenisKelamin = scanner.nextBoolean();                  scanner.nextLine(); // ambil sisa new line                  System.out.print("Masukkan Alamat Baru: ");                  String alamat = scanner.nextLine();                  System.out.print("Masukkan Nomor Telepon Baru: ");                  String noTelp = scanner.nextLine();                  System.out.print("Masukkan Spesialisasi Baru: ");                  String spesialisasi = scanner.nextLine();                  dokter = new Dokter28(id, nama, jenisKelamin, alamat, noTelp, spesialisasi);                  System.out.println("Data dokter berhasil diupdate.");                  return;              }          }          System.out.println("Dokter dengan ID " + id + " tidak ditemukan.");      }      public static void updatePasien(Scanner scanner) {          System.out.print("Masukkan ID Pasien yang ingin diupdate: ");          int id = scanner.nextInt();          scanner.nextLine(); // ambil sisa new line          for (Pasien28 pasien : daftarPasien) {              if (pasien.getId() == id) {                  System.out.print("Masukkan Nama Baru: ");                  String nama = scanner.nextLine();                  System.out.print("Masukkan Jenis Kelamin Baru (true untuk laki-laki, false untuk perempuan): ");                  boolean jenisKelamin = scanner.nextBoolean();                  scanner.nextLine(); // ambil sisa new line                  System.out.print("Masukkan Alamat Baru: ");                  String alamat = scanner.nextLine();                  System.out.print("Masukkan Nomor Telepon Baru: ");                  String noTelp = scanner.nextLine();                  pasien = new Pasien28(id, nama, jenisKelamin, alamat, noTelp);                  System.out.println("Data pasien berhasil diupdate.");                  return;              }          }          System.out.println("Pasien dengan ID " + id + " tidak ditemukan.");      }  } |