

SS Program Posttest 6
Muhammad Khemal Arsenio Bohari (2309106098)
C1'23

```
KlinikAntrian.java > KlinikAntrian
1  import java.util.ArrayList;
2  import java.util.Scanner;
3
4  public final class KlinikAntrian {
5      private static ArrayList<Pasien> daftarPasien = new ArrayList<>();
6      private static Scanner scanner = new Scanner(System.in);
7      public static int totalPasien = 0; // static variable
8
9      public static void main(String[] args) {
10         while (true) {
11             try {
12                 System.out.println(x:"\n== Sistem Klinik ==");
13                 System.out.println(x:"1. Masuk sebagai Admin");
14                 System.out.println(x:"2. Masuk sebagai Pasien");
15                 System.out.println(x:"3. Keluar");
16                 System.out.print(s:"Pilih menu: ");
17                 int pilihan = scanner.nextInt();
18                 scanner.nextLine();
19
20                 switch (pilihan) {
21                     case 1:
22                         menuAdmin();
23                         break;
24                     case 2:
25                         KlinikPasien.menuPasien(daftarPasien);
26                         break;
27                     case 3:
28                         System.out.println(x:"Terima kasih! Program selesai.");
29                         return;
30                     default:
31                         System.out.println(x:"Pilihan tidak valid. Silakan coba lagi.");
32                 }
33             } catch (Exception e) {
34                 System.out.println("Terjadi kesalahan: " + e.getMessage());
35                 scanner.nextLine();
36             }
37         }
38     }
39
40     private static void menuAdmin() {
41         while (true) {
42             try {
43                 System.out.println(x:"\n== Menu Admin Klinik ==");
44                 System.out.println(x:"1. Tambah Pasien Umum");
45                 System.out.println(x:"2. Tambah Pasien BPJS");
46                 System.out.println(x:"3. Lihat Daftar Pasien");
47                 System.out.println(x:"4. Ubah Data Pasien");
48                 System.out.println(x:"5. Hapus Pasien");
49                 System.out.println(x:"6. Tampilkan Total Pasien");
50                 System.out.println(x:"7. Kembali ke Menu Utama");
51                 System.out.print(s:"Pilih menu: ");
52                 int pilihan = scanner.nextInt();
53                 scanner.nextLine();
54
55                 switch (pilihan) {
56                     case 1:
57                         tambahPasienUmum();
58                         break;
59                     case 2:
60                         tambahPasienBPJS();
61                         break;
62                     case 3:
63                         lihatDaftarPasien();
64                         break;
65                     case 4:
66                         ubahDataPasien();
67                         break;
68                     case 5:
69                         hapusPasien();
70                         break;
69                     case 6:
70                         tampilkanTotalPasien();
71                         break;
72                     case 7:
73                         return;
74                     default:
75                         System.out.println(x:"Pilihan tidak valid. Silakan coba lagi.");
76                 }
77             } catch (Exception e) {
78                 System.out.println("Terjadi kesalahan: " + e.getMessage());
79                 scanner.nextLine();
80             }
81         }
82     }
83
84     private static void tambahPasienUmum() {
85         // ...
86     }
87
88     private static void tambahPasienBPJS() {
89         // ...
90     }
91
92     private static void lihatDaftarPasien() {
93         // ...
94     }
95
96     private static void ubahDataPasien() {
97         // ...
98     }
99
100    private static void hapusPasien() {
101        // ...
102    }
103
104    private static void tampilkanTotalPasien() {
105        // ...
106    }
107 }
```

Activate Windows
Go to Settings to activate Windows.

Activate Windows
Go to Settings to activate Windows.

```

        tambahPasienBPJS(),
        break;
    case 3:
        lihatPasien();
        break;
    case 4:
        ubahPasien();
        break;
    case 5:
        hapusPasien();
        break;
    case 6:
        tampilkanTotalPasien();
        break;
    case 7:
        return;
    default:
        System.out.println(x:"Pilihan tidak valid. Silakan coba lagi.");
    }
} catch (Exception e) {
    System.out.println("Terjadi kesalahan: " + e.getMessage());
    scanner.nextLine();
}
}
}

```

```

private static void tambahPasienUmum() {
    try {
        System.out.print(s:"Masukkan nama pasien: ");
        String nama = scanner.nextLine();

```

Activate Windows
Go to Settings to activate Windows.

```

        System.out.print(s:"Masukkan umur pasien: ");
        int umur = scanner.nextInt(); scanner.nextLine();
        System.out.print(s:"Masukkan keluhan pasien: ");
        String keluhan = scanner.nextLine();

        Pasien pasien = new PasienUmum(nama, umur, keluhan);
        daftarPasien.add(pasien);
        pasien.daftar();
        System.out.println(x:"Pasien umum berhasil ditambahkan!");
    } catch (Exception e) {
        System.out.println("Terjadi kesalahan saat menambahkan pasien: " + e.getMessage());
    }
}

```

```

private static void tambahPasienBPJS() {
    try {
        System.out.print(s:"Masukkan nama pasien: ");
        String nama = scanner.nextLine();
        System.out.print(s:"Masukkan umur pasien: ");
        int umur = scanner.nextInt(); scanner.nextLine();
        System.out.print(s:"Masukkan keluhan pasien: ");
        String keluhan = scanner.nextLine();
        System.out.print(s:"Masukkan nomor BPJS: ");
        String nomorBPJS = scanner.nextLine();

        Pasien pasien = new PasienBPJS(nama, umur, keluhan, nomorBPJS);
        daftarPasien.add(pasien);
        pasien.daftar();
        System.out.println(x:"Pasien BPJS berhasil ditambahkan!");
    } catch (Exception e) {

```

Activate Windows
Go to Settings to activate Windows.

```

    } catch (Exception e) {
        System.out.println("Terjadi kesalahan saat menambahkan pasien BPJS: " + e.getMessage());
    }
}

private static void lihatPasien() {
    if (daftarPasien.isEmpty()) {
        System.out.println(x:"Tidak ada pasien dalam daftar.");
        return;
    }
    System.out.println(x:"\nDaftar Pasien:");
    for (int i = 0; i < daftarPasien.size(); i++) {
        Pasien p = daftarPasien.get(i);
        System.out.println((i + 1) + ". " + p);
        p.infoTambahan();
    }
}

private static void ubahPasien() {
    lihatPasien();
    if (daftarPasien.isEmpty()) return;
    try {
        System.out.print(s:"Pilih nomor pasien yang ingin diubah: ");
        int index = scanner.nextInt() - 1; scanner.nextLine();
        if (index < 0 || index >= daftarPasien.size()) {
            System.out.println(x:"Nomor pasien tidak valid.");
            return;
        }
        Pasien pasien = daftarPasien.get(index);
        System.out.print(s:"Masukkan nama baru: ");
        pasien.setNama(scanner.nextLine());
        System.out.print(s:"Masukkan umur baru: ");
        pasien.setUmur(scanner.nextInt()); scanner.nextLine();
        System.out.print(s:"Masukkan keluhan baru: ");
        pasien.setKeluhan(scanner.nextLine());
        if (pasien instanceof PasienBPJS) {
            System.out.print(s:"Masukkan nomor BPJS baru: ");
            ((PasienBPJS) pasien).setNomorBPJS(scanner.nextLine());
        }
        System.out.println(x:"Data pasien berhasil diubah!");
    } catch (Exception e) {
        System.out.println("Terjadi kesalahan saat mengubah data pasien: " + e.getMessage());
    }
}

private static void hapusPasien() {
    lihatPasien();
    if (daftarPasien.isEmpty()) return;
    try {
        System.out.print(s:"Pilih nomor pasien yang ingin dihapus: ");
        int index = scanner.nextInt() - 1; scanner.nextLine();
        if (index < 0 || index >= daftarPasien.size()) {
            System.out.println(x:"Nomor pasien tidak valid.");
            return;
        }
        Pasien pasien = daftarPasien.get(index);
        pasien.batalDaftar();
        daftarPasien.remove(index);
        totalPasien--; // static decrement
        System.out.println(x:"Pasien berhasil dihapus!");
    } catch (Exception e) {
        System.out.println("Terjadi kesalahan saat menghapus pasien: " + e.getMessage());
    }
}

public static void tampilkanTotalPasien() {
    System.out.println("Total pasien terdaftar: " + totalPasien);
}

```

```
KlinikPasien.java > ...
1  import java.util.ArrayList;
2  import java.util.Scanner;
3
4  public class KlinikPasien {
5      private static Scanner scanner = new Scanner(System.in);
6
7      public static void menuPasien(ArrayList<Pasien> daftarPasien) {
8          while (true) {
9              System.out.println(x:"\\n== Menu Pasien Klinik ==");
10             System.out.println(x:"1. Daftar Sebagai Pasien Baru");
11             System.out.println(x:"2. Lihat Daftar Pasien");
12             System.out.println(x:"3. Informasi Klinik");
13             System.out.println(x:"4. Batal Antrian");
14             System.out.println(x:"5. Kembali ke Menu Utama");
15             System.out.print(s:"Pilih menu: ");
16             int pilihan = scanner.nextInt();
17             scanner.nextLine();
18
19             switch (pilihan) {
20                 case 1:
21                     daftarPasienBaru(daftarPasien);
22                     break;
23                 case 2:
24                     lihatAntrian(daftarPasien);
25                     break;
26                 case 3:
27                     System.out.println(x:"\\nInformasi Klinik:");
28                     System.out.println(x:"Jam buka: 08:00 - 22:00");
29                     System.out.println(x:"Dokter: Dr. Khemal, Dr. Luhut, Dr. Opik");
30                     System.out.println(x:"Hari pelayanan: Senin - Jumat");
31                     break;
32                 case 4:
33                     batalAntrian(daftarPasien);
34                     break;
35                 case 5:
36                     return;
37                 default:
38                     System.out.println(x:"Pilihan tidak valid. Silakan coba lagi.");
39             }
40         }
41     }
42
43     private static void daftarPasienBaru(ArrayList<Pasien> daftarPasien) {
44         System.out.println(x:"Pilih tipe pasien:");
45         System.out.println(x:"1. Umum");
46         System.out.println(x:"2. BPJS");
47         System.out.print(s:"Pilihan: ");
48         int tipe = scanner.nextInt();
49         scanner.nextLine();
50
51         System.out.print(s:"Masukkan nama pasien: ");
52         String nama = scanner.nextLine();
53         System.out.print(s:"Masukkan umur pasien: ");
54         int umur = scanner.nextInt();
55         scanner.nextLine();
56         System.out.print(s:"Masukkan keluhan pasien: ");
57         String keluhan = scanner.nextLine();
58
59         Pasien pasien;
60         if (tipe == 1) {
61             pasien = new PasienUmum(nama, umur, keluhan);
62         } else if (tipe == 2) {
63             pasien = new PasienBPJS(nama, umur, keluhan);
64         }
65         daftarPasien.add(pasien);
66     }
67
68     private static void lihatAntrian(ArrayList<Pasien> daftarPasien) {
69         if (daftarPasien.isEmpty()) {
70             System.out.println(x:"Tidak ada pasien dalam antrian.");
71         } else {
72             System.out.println(x:"\\nDaftar Pasien dalam Antrian:");
73             for (Pasien pasien : daftarPasien) {
74                 System.out.println(x:"Nama: " + pasien.getNama() + ", Umur: " + pasien.getUmur() + ", Keluhan: " + pasien.getKeluhan());
75             }
76         }
77     }
78
79     private static void batalAntrian(ArrayList<Pasien> daftarPasien) {
80         if (daftarPasien.isEmpty()) {
81             System.out.println(x:"Tidak ada pasien dalam antrian untuk dibatalkan.");
82         } else {
83             System.out.println(x:"\\nDaftar Pasien dalam Antrian:");
84             for (Pasien pasien : daftarPasien) {
85                 System.out.println(x:"Nama: " + pasien.getNama() + ", Umur: " + pasien.getUmur() + ", Keluhan: " + pasien.getKeluhan());
86             }
87             System.out.print(s:"Pilih nomor pasien yang akan dibatalkan: ");
88             int index = scanner.nextInt();
89             scanner.nextLine();
90             if (index < 0 || index >= daftarPasien.size()) {
91                 System.out.println(x:"Nomor pasien tidak valid.");
92             } else {
93                 daftarPasien.remove(index);
94                 System.out.println(x:"Pasien dengan nomor " + index + " telah dibatalkan dari antrian.");
95             }
96         }
97     }
98
99     private static void informasiKlinik() {
100         System.out.println(x:"\\nInformasi Klinik:");
101         System.out.println(x:"Jam buka: 08:00 - 22:00");
102         System.out.println(x:"Dokter: Dr. Khemal, Dr. Luhut, Dr. Opik");
103         System.out.println(x:"Hari pelayanan: Senin - Jumat");
104     }
105
106     private static void kembaliKeMenuUtama() {
107         System.out.println(x:"\\nKembali ke Menu Utama.");
108     }
109 }
```

```

private static void daftarPasien(ArrayList<Pasien> daftarPasien) {
    System.out.print(s:"Masukkan nomor BPJS: ");
    String noBPJS = scanner.nextLine();
    pasien = new PasienBPJS(nama, umur, keluhan, noBPJS);
} else {
    System.out.println(x:"Tipe tidak valid, pendaftaran dibatalkan.");
    return;
}

daftarPasien.add(pasien);
pasien.daftar();
pasien.infoTambahan();
}

private static void lihatAntrian(ArrayList<Pasien> daftarPasien) {
    System.out.println(x:"\nDaftar Pasien di Antrian:");
    if (daftarPasien.isEmpty()) {
        System.out.println(x:"Tidak ada pasien dalam antrian.");
    } else {
        for (int i = 0; i < daftarPasien.size(); i++) {
            System.out.println((i + 1) + ". " + daftarPasien.get(i));
            daftarPasien.get(i).infoTambahan();
        }
    }
}
}

```

```

private static void batalAntrian(ArrayList<Pasien> daftarPasien) {
    System.out.print(s:"Masukkan nama pasien yang ingin membatalkan antrian: ");
    String nama = scanner.nextLine();
    boolean found = false;
    for (int i = 0; i < daftarPasien.size(); i++) {
        for (int i = 0; i < daftarPasien.size(); i++) {
            if (daftarPasien.get(i).getNama().equalsIgnoreCase(nama)) {
                daftarPasien.get(i).batalDaftar();
                daftarPasien.remove(i);
                KlinikAntrian.totalPasien--; // static decrement
                System.out.println("Antrian pasien " + nama + " berhasil dibatalkan.");
                found = true;
                break;
            }
        }
    }
    if (!found) {
        System.out.println(x:"Pasien dengan nama tersebut tidak ditemukan.");
    }
}
}

```

Activate Windows
Go to Settings to activate Windows.

Pasien.java > ...

```

1 public abstract class Pasien implements Pendaftaran {
2     private final String tipePasien = "Umum";
3     private String nama;
4     private int umur;
5     private String keluhan;
6
7     public Pasien(String nama, int umur, String keluhan) {
8         try {
9             if (nama == null || nama.trim().isEmpty()) {
10                 throw new IllegalArgumentException(s:"Nama pasien tidak boleh kosong!");
11             }
12             this.nama = nama;
13             setUmur(umur);
14             this.keluhan = keluhan;
15             KlinikAntrian.totalPasien++; // static variable increment
16         } catch (Exception e) {
17             System.out.println("Error: " + e.getMessage());
18         }
19     }
20
21     public String getTypePasien() {
22         return tipePasien;
23     }
24
25     public String getNama() {
26         return nama;
27     }
28
29     public void setName(String nama) {
30         this.nama = nama;
31     }
32

```

Activate Windows
Go to Settings to activate Windows.


```

    public int getUmur() {
        return umur;
    }

    public void setUmur(int umur) {
        if (umur > 0) {
            this.umur = umur;
        } else {
            System.out.println(x:"Umur harus lebih dari 0!");
        }
    }

    public String getKeluhan() {
        return keluhan;
    }

    public void setKeluhan(String keluhan) {
        this.keluhan = keluhan;
    }

    @Override
    public void daftar() {
        System.out.println(nama + " berhasil mendaftar sebagai " + tipePasien + ".");
    }

    @Override
    public void batalDaftar() {
        System.out.println(nama + " membatalkan pendaftaran.");
    }
}

```

Activate Windows
Go to Settings to activate Windows.

```

61 }
62
63 public abstract void infoTambahan();
64
65 public final void tampilkanIdentitas() {
66     System.out.println("Pasien Klinik - Tipe: " + tipePasien);
67 }
68
69 @Override
70 public String toString() {
71     return "Nama: " + nama + ", Umur: " + umur + ", Keluhan: " + keluhan;
72 }
73 }
74

```

PasienBPJS.java > ...

```

1 public class PasienBPJS extends Pasien {
2     private String nomorBPJS;
3
4     public PasienBPJS(String nama, int umur, String keluhan, String nomorBPJS) {
5         super(nama, umur, keluhan);
6         this.nomorBPJS = nomorBPJS;
7     }
8
9     public void setNomorBPJS(String nomorBPJS) {
10         this.nomorBPJS = nomorBPJS;
11     }
12
13     public String getNomorBPJS() {
14         return nomorBPJS;
15     }
16
17     @Override
18     public void infoTambahan() {
19         System.out.println("Jenis Pasien: BPJS. Nomor: " + nomorBPJS);
20     }
21 }
22


```

PasienUmum.java > ...

```

1 public class PasienUmum extends Pasien {
2     public PasienUmum(String nama, int umur, String keluhan) {
3         super(nama, umur, keluhan);
4     }
5
6     @Override
7     public void infoTambahan() {
8         System.out.println(x:"Pasien umum tidak memiliki asuransi khusus.");
9     }
10 }
11

```

 Pendaftaran.java > ...

```
1  public interface Pendaftaran {  
2      void daftar();  
3      void batalDaftar();  
4  }  
5
```