

POSTTEST 6

NAMA : NUR ALISA

NIM : 2209106089

SOURCE CODE PROGRAM

Class App

```
POSTTEST6 > src > Appjava > App > main(String[])
1 import java.io.FileInputStream;
2 import java.io.FileOutputStream;
3 import java.io.IOException;
4 import java.io.ObjectInputStream;
5 import java.io.ObjectOutputStream;
6 import java.util.ArrayList;
7 import java.util.InputMismatchException;
8 import java.util.Scanner;
9
10 public class App implements PengelolaanData {
11     private static Scanner scanner = new Scanner(System.in);
12     private static ArrayList<Mebel> daftarMebel = new ArrayList<>();
13     private static final String FILE_NAME = "data_mebel.txt";
14     Run | Debug
15     public static void main(String[] args) {
16         boolean sedangBerjalan = true;
17
18         while (sedangBerjalan) {
19             System.out.println(x:"***** MEBEL KREATIF *****");
20             System.out.println(x:"|| >> [1] Tampilkan Data Furnitur << ||");
21             System.out.println(x:"|| >> [2] Tambah Furnitur Baru << ||");
22             System.out.println(x:"|| >> [3] Ubah Data Furnitur << ||");
23             System.out.println(x:"|| >> [4] Hapus Furnitur << ||");
24             System.out.println(x:"|| >> [5] Keluar << ||");
25             System.out.println(x:"*****");
26             System.out.print(s:"Pilih Menu (1-5): ");
27             try {
28                 int pilihan = scanner.nextInt();
29                 scanner.nextLine();
30
31                 switch (pilihan) {
32                     case 1:
33                         tampilkanDataFurnitur();
34                         break;
35                     case 2:
36                         tambahFurniturBaru();
37                         break;
38                     case 3:
39                         ubahDataFurnitur();
40                         break;
```

```
case 4:
    hapusFurnitur();
    break;
case 5:
    System.out.println(x:"*****");
    System.out.println(x:" TERIMA KASIH ");
    System.out.println(x:"*****");
    sedangBerjalan = false;
    break;
default:
    System.out.println(x:"Pilihan tidak valid!");
}
} catch (InputMismatchException e) {
    System.out.println(x:"Input tidak valid. Silakan masukkan angka.");
    scanner.nextLine();
}
}

private static void tampilkanDataFurnitur() {
    if (daftarMebel.isEmpty()) {
        System.out.println();
        System.out.println(x:"Data furnitur masih kosong!");
        return;
    }

    System.out.println(x:"***** Pilih Furnitur *****");
    System.out.println(x:"|| >> [1] Meja << ||");
    System.out.println(x:"|| >> [2] Kursi << ||");
    System.out.println(x:"|| >> [3] Lemari << ||");
    System.out.println(x:"|| >> [4] Kasur << ||");
    System.out.println(x:"|| >> [5] Kembali << ||");
    System.out.println(x:"*****");
    System.out.print(s:"Pilih Menu (1-5): ");
```

```
int pilihanKelas = scanner.nextInt();
scanner.nextLine();

switch (pilihanKelas) {
    case 1:
        tampilkanDataKelas(kelas:Meja.class);
        break;
    case 2:
        tampilkanDataKelas(kelas:Kursi.class);
        break;
    case 3:
        tampilkanDataKelas(kelas:Lemari.class);
        break;
    case 4:
        tampilkanDataKelas(kelas:Kasur.class);
        break;
    case 5:
        System.out.println();
        break;
    default:
        System.out.println(x:"Pilihan kelas tidak valid!");
        break;
}

private static void tampilkanDataKelas(class? extends Mebel? kelas) {
    System.out.println("Data " + kelas.getSimpleName() + ":");

    System.out.println(x:"*****");
    System.out.printf(format:"%5s%-10s%-10s%-10s%-10s\n", ...args:"ID", "Nama", "Material", "Harga", "Stok", "Kelas");
    System.out.println(x:"*****");
```

```
for (Mebel mebel : daftarMebel) {
    if (kelas.isInstance(mebel)) {
        System.out.printf(format:"%-5s%-10s%-10s%-10s%-10s\n", mebel.getId(), mebel.getNama(),
            mebel.getMaterial(), mebel.getHarga(),
            mebel.getStok(), kelas.getSimpleName());
        System.out.println(x:"-----");
    }
}

private static void tambahFurniturBaru() {
    System.out.println(x:"***** Tambah Furnitur *****");
    System.out.println(x:"|| >> [1] Meja << ||");
    System.out.println(x:"|| >> [2] Kursi << ||");
    System.out.println(x:"|| >> [3] Lemari << ||");
    System.out.println(x:"|| >> [4] Kasur << ||");
    System.out.println(x:"|| >> [5] Kembali << ||");
    System.out.println(x:"*****");
    System.out.print(s:"Pilih Menu (1-5): ");

    int pilihan = scanner.nextInt();
    scanner.nextLine();

    String nama, material;
    int harga, stok;

    switch (pilihan) {
        case 1:
            System.out.print(s:"Masukkan Nama Meja: ");
            nama = scanner.nextLine();
            System.out.print(s:"Masukkan Material Meja: ");
            material = scanner.nextLine();
            System.out.print(s:"Masukkan Harga Meja: ");
            harga = scanner.nextInt();
```

```

        System.out.print(s:"Masukkan Harga Meja: ");
        harga = scanner.nextInt();
        System.out.print(s:"Masukkan Stok Meja: ");
        stok = scanner.nextInt();
        scanner.nextLine();
        System.out.print(s:"Masukkan Jumlah Kaki Meja: ");
        int jumlahKakiMeja = scanner.nextInt();
        scanner.nextLine();
        daftarMebel.add(new Meja(nama, material, harga, stok, jumlahKakiMeja));
        System.out.println(x:"Meja berhasil ditambahkan!");
        break;
    case 2:
        System.out.print(s:"Masukkan Nama Kursi: ");
        nama = scanner.nextLine();
        System.out.print(s:"Masukkan Material Kursi: ");
        material = scanner.nextLine();
        System.out.print(s:"Masukkan Harga Kursi: ");
        harga = scanner.nextInt();
        System.out.print(s:"Masukkan Stok Kursi: ");
        stok = scanner.nextInt();
        scanner.nextLine();
        System.out.print(s:"Masukkan Jenis Kursi: ");
        String jenisKursi = scanner.nextLine();
        daftarMebel.add(new Kursi(nama, material, harga, stok, jenisKursi));
        System.out.println(x:"Kursi berhasil ditambahkan!");
        break;
    case 3:
        System.out.print(s:"Masukkan Nama Lemari: ");
        nama = scanner.nextLine();
        System.out.print(s:"Masukkan Material Lemari: ");
        material = scanner.nextLine();
        System.out.print(s:"Masukkan Harga Lemari: ");
        harga = scanner.nextInt();
        System.out.print(s:"Masukkan Stok Lemari: ");
        stok = scanner.nextInt();

```

```

        case 4:
            System.out.print(s:"Masukkan Nama Kasur: ");
            nama = scanner.nextLine();
            System.out.print(s:"Masukkan Material Kasur: ");
            material = scanner.nextLine();
            System.out.print(s:"Masukkan Harga Kasur: ");
            harga = scanner.nextInt();
            System.out.print(s:"Masukkan Stok Kasur: ");
            stok = scanner.nextInt();
            daftarMebel.add(new Kasur(nama, material, harga, stok));
            System.out.println(x:"Kasur berhasil ditambahkan!");
            break;
        case 5:
            System.out.println();
            break;
        default:
            System.out.println(x:"Pilihan tidak valid!");
    }
}

private static void ubahDataFurnitur() {
    if (daftarMebel.isEmpty()) {
        System.out.println(x:"Data furnitur masih kosong!");
        return;
    }

    tampilkanDataFurnitur();

    System.out.print("Masukkan nomor furnitur yang ingin diubah (1-" + daftarMebel.size() + "): ");
    int pilihan = scanner.nextInt() - 1;

    if (pilihan >= 0 && pilihan < daftarMebel.size()) {
        Mebel furnitur = daftarMebel.get(pilihan);

```

```

        System.out.print("Masukkan nomor furnitur yang ingin diubah (1-" + daftarMebel.size() + "): ");
        int pilihan = scanner.nextInt() - 1;

        if (pilihan >= 0 && pilihan < daftarMebel.size()) {
            Mebel furnitur = daftarMebel.get(pilihan);

            System.out.println("**Ubah data " + furnitur.getNama() + "**");
            System.out.print("Nama baru (" + furnitur.getNama() + "): ");
            String namaBaru = scanner.next();
            if (!namaBaru.isEmpty()) {
                furnitur.setNama(namaBaru);
            }

            System.out.print("Material baru (" + furnitur.getMaterial() + "): ");
            String materialBaru = scanner.next();
            if (!materialBaru.isEmpty()) {
                furnitur.setMaterial(materialBaru);
            }

            System.out.print("Harga baru (Rp" + furnitur.getHarga() + "): ");
            int hargaBaru = scanner.nextInt();
            if (hargaBaru > 0) {
                furnitur.setHarga(hargaBaru);
            }

            System.out.print("Stok baru (" + furnitur.getStok() + "): ");
            int stokBaru = scanner.nextInt();
            if (stokBaru >= 0) {
                furnitur.setStok(stokBaru);
            }

            System.out.println(x:"Data furnitur berhasil diubah!");
        } else {
            System.out.println(x:"Nomor furnitur tidak valid!");
        }
    }
}

```

```

private static void hapusFurnitur() {
    if (daftarMebel.isEmpty()) {
        System.out.println(x:"Data furnitur masih kosong!");
        return;
    }

    tampilkanDataFurnitur();

    System.out.print("Masukkan nomor furnitur yang ingin dihapus (1-" + daftarMebel.size() + "): ");
    int pilihan = scanner.nextInt() - 1;

    if (pilihan >= 0 && pilihan < daftarMebel.size()) {
        daftarMebel.remove(pilihan);
        System.out.println(x:"Furnitur berhasil dihapus!");
    } else {
        System.out.println(x:"Nomor furnitur tidak valid!");
    }
}

```

```

public void simpanData() {
    try (ObjectOutputStream outputStream = new ObjectOutputStream(new FileOutputStream(FILE_NAME))) {
        outputStream.writeObject(daftarMebel);
        System.out.println(x:"Data berhasil disimpan.");
    } catch (IOException e) {
        tanganiKesalahan(e);
    }
}

@Override
public void muatData() {
    try (ObjectInputStream inputStream = new ObjectInputStream(new FileInputStream(FILE_NAME))) {
        daftarMebel = (ArrayList<Mebel>) inputStream.readObject();
        System.out.println(x:"Data berhasil dimuat.");
    } catch (IOException | ClassNotFoundException e) {
        tanganiKesalahan(e);
    }
}

public static void tanganiKesalahan(Exception e) {
    System.out.println("Terjadi kesalahan: " + e.getMessage());
}
}

```

CLASS MEBEL/ ABSTRAC CLASS

```
public abstract class Mebel {
    private static int counter = 1;
    protected final int id;
    protected String nama;
    protected String material;
    protected int harga;
    protected int stok;

    public Mebel(String nama, String material, int harga, int stok) {
        this.id = counter++;
        this.nama = nama;
        this.material = material;
        this.harga = harga;
        this.stok = stok;
    }

    public Mebel(String nama, String material, int harga) {
        this(nama, material, harga, stok:0);
    }

    public void setStok() {
        this.stok = 0;
    }

    public void setStok(int stok) {
        this.stok = stok;
    }

    public int getId() {
        return id;
    }

    public String getNama() {
        return nama;
    }
}
```

```
public void setNama(String nama) {
    this.nama = nama;
}

public String getMaterial() {
    return material;
}

public void setMaterial(String material) {
    this.material = material;
}

public int getHarga() {
    return harga;
}

public void setHarga(int harga) {
    this.harga = harga;
}

public int getStok() {
    return stok;
}

public abstract void hitungJumlah();

@Override
public String toString() {
    return "ID: " + id + ", Nama: " + nama + ", Material: " + material + ", Harga: Rp" + harga + ", Stok: " + stok;
}
```

CLASS KURSI

```
public class Kursi extends Mebel {
    private final String jenisKursi;

    public Kursi(String nama, String material, int harga, int stok, String jenisKursi) {
        super(nama, material, harga, stok);
        this.jenisKursi = jenisKursi;
    }

    public String getJenisKursi() {
        return jenisKursi;
    }

    @Override
    public void hitungJumlah() {
    }

    @Override
    public String toString() {
        return super.toString() + ", Jenis Kursi: " + jenisKursi;
    }
}
```

CLASS MEJA

```
public class Meja extends Mebel {
    private final int jumlahKaki;

    public Meja(String nama, String material, int harga, int stok, int jumlahKaki) {
        super(nama, material, harga, stok);
        this.jumlahKaki = jumlahKaki;
    }

    public int getJumlahKaki() {
        return jumlahKaki;
    }

    @Override
    public void hitungJumlah() {
    }

    @Override
    public final void setStok() {
        this.stok = 0;
    }

    @Override
    public final void setStok(int stok) {
        this.stok = stok;
    }

    @Override
    public String toString() {
        return super.toString() + ", Jumlah Kaki: " + jumlahKaki;
    }
}
```

CLASS LEMARI

```
public class Lemari extends Mebel {
    private final int jumlahPintu;

    public Lemari(String nama, String material, int harga, int stok, int jumlahPintu) {
        super(nama, material, harga, stok);
        this.jumlahPintu = jumlahPintu;
    }

    public int getJumlahPintu() {
        return jumlahPintu;
    }

    @Override
    public void hitungJumlah() {

    }

    @Override
    public String toString() {
        return super.toString() + ", Jumlah Pintu: " + jumlahPintu;
    }
}
```

CLASS KASUR

```
public class Kasur extends Mebel {

    public Kasur(String nama, String material, int harga, int stok) {
        super(nama, material, harga, stok);
    }

    @Override
    public void hitungJumlah() {

    }
}
```

CLASS PENGELOLAAN DATA

```
POSTTEST6 > src > PengelolaanData.java > ...
1  public interface PengelolaanData {
2      void simpanData();
3      void muatData();
4  }
```