

Nama : Yunan Faila Sofi

NIM : H1D024033

ShiftLama : G

ShiftBaru : C

Class BarangMusik

package pert2;

```
public class BarangMusik { // Deklarasikan variabel/state yang diperlukan di sini
private String kodeBarang;
private String namaAlatMusik;
private double hargaSatuan;
private int jumlahStok;
```

```
// --- Constructor (Method Overloading) ---
```

```
// Cara ketiga: Constructor Data Lengkap (Kode, Nama, Harga, Stok)
public BarangMusik(String kode, String nama, double harga, int stok) {
    this.kodeBarang = kode;
    this.namaAlatMusik = nama;
    this.hargaSatuan = harga;
    this.jumlahStok = stok;
}
```

```
// Cara kedua: Constructor Kode, Nama, dan Harga
// Stok otomatis diset 0.
public BarangMusik(String kode, String nama, double harga) {
    // Memanggil constructor lengkap, dengan stok diset 0
    this(kode, nama, harga, 0);
}
```

```
// Cara pertama: Constructor Hanya Kode dan Nama
// Harga otomatis diset 0.0 dan Stok otomatis diset 0.
public BarangMusik(String kode, String nama) {
    // Memanggil constructor harga (yang kemudian memanggil constructor lengkap)
    this(kode, nama, 0.0);
}
```

```
// --- Kemampuan Sistem ---
```

```

/**
 * Mengubah harga barang.
 * @param hargaBaru Harga satuan baru.
 */
public void ubahHarga(double hargaBaru) {
    if (hargaBaru >= 0) {
        this.hargaSatuan = hargaBaru;
        // System.out.println("Harga " + this.namaAlatMusik + " berhasil diubah menjadi Rp " +
hargaBaru);
    } else {
        System.out.println("Harga tidak valid.");
    }
}

/**
 * Menambah stok barang.
 * @param jumlah Jumlah barang yang ditambahkan.
 */
public void tambahStok(int jumlah) {
    if (jumlah > 0) {
        this.jumlahStok += jumlah;
        // System.out.println("Stok " + this.namaAlatMusik + " bertambah " + jumlah + " unit. Total
stok: " + this.jumlahStok);
    } else {
        System.out.println("Jumlah penambahan stok harus positif.");
    }
}

/**
 * Menampilkan informasi lengkap barang ke konsol.
 * Format: Alat musik [nama] | Kode: [kode] | Harga: Rp [harga] | Stok: [stok] unit
 */
public void tampilInfo() {
    System.out.println(
        "Alat musik " + this.namaAlatMusik +
        " | Kode: " + this.kodeBarang +
        " | Harga: Rp " + this.hargaSatuan +
        " | Stok: " + this.jumlahStok + " unit"
    );
}

```

```
}  
}  
}
```

I. Alur Kerja Program (Workflow)

Alur logika program berjalan secara berurutan (*sequential*) dari class main (UjiBarang), dengan fokus pada penggunaan **Constructor Overloading**:

1. **Start (Inisialisasi):** Program dimulai dari method main di class UjiBarang.
2. **Instansiasi Objek 1 (Gitar Akustik):**
 - Program membuat objek gitarAkustik menggunakan **Constructor Cara Pertama** (hanya Kode dan Nama).
 - *Proses Internal (Constructor Chaining):* Constructor ini memanggil constructor kedua (tambah harga 0.0), yang kemudian memanggil constructor ketiga (tambah stok 0). Akhirnya, Gitar terbentuk dengan Harga 0 dan Stok 0.
3. **Modifikasi State Objek 1:**
 - Memanggil ubahHarga(1500000.0): Mengubah harga gitar dari 0 menjadi 1.500.000.
 - Memanggil tambahStok(5): Menambah stok gitar dari 0 menjadi 5.
4. **Instansiasi Objek 2 (Drum Set):**
 - Program membuat objek drumSet menggunakan **Constructor Cara Ketiga** (Data Lengkap).
 - Data Kode, Nama, Harga (8.500.000), dan Stok (3) langsung diisi saat pembuatan objek.
5. **Output Informasi:**
 - Mencetak header "=== INVENTARIS TOKO NADAKITA ===".
 - Memanggil gitarAkustik.tampilInfo() untuk menampilkan data terkini gitar.
 - Memanggil drumSet.tampilInfo() untuk menampilkan data drum.
6. **End:** Program selesai.

II. Fungsi / Method yang Digunakan

Berikut adalah rincian fungsi (method) dalam class BarangMusik:

1. **Constructor Overloading (3 Variasi)**
 - **Jenis:** Special Method (Inisialisasi).

- **Fungsi:** Memberikan fleksibilitas cara pembuatan objek.
 - **BarangMusik(kode, nama):** Untuk barang yang belum tahu harga dan stoknya.
 - **BarangMusik(kode, nama, harga):** Untuk barang yang belum masuk gudang (stok 0).
 - **BarangMusik(kode, nama, harga, stok):** Untuk pencatatan data lengkap.
- **Fitur:** Menggunakan keyword `this(...)` untuk meminimalisir duplikasi kode antar constructor.

2. **ubahHarga(double hargaBaru)**

- **Jenis:** Setter / Mutator.
- **Logika:** Melakukan validasi `if (hargaBaru >= 0)`. Jika valid, nilai atribut `hargaSatuan` diperbarui.
- **Tujuan:** Memungkinkan perubahan harga seiring waktu (enkapsulasi).

3. **tambahStok(int jumlah)**

- **Jenis:** Setter / Mutator (Akumulatif).
- **Logika:** Melakukan validasi `if (jumlah > 0)`. Jika valid, menambahkan nilai ke stok yang ada (`this.jumlahStok += jumlah`).
- **Tujuan:** Menangani penambahan barang masuk (restock).

4. **tampilInfo()**

- **Jenis:** Prosedur (Void).
- **Logika:** Menggabungkan string dan variabel (concatenation) lalu mencetaknya ke konsol.
- **Tujuan:** Menampilkan laporan status inventaris barang secara rapi.

Class UjiBarang

`package pert2;`

```
public class UjiBarang {
    public static void main(String[] args) {
```

```
        // 1. Barang Pertama: Gitar Akustik (Cara pertama: hanya kode dan nama)
        // Kode: "GTR-001", Nama: "Gitar Akustik Yamaha"
```

```

BarangMusik gitarAkustik = new BarangMusik("GTR-001", "Gitar Akustik Yamaha");

// Atur harga menjadi Rp 1.500.000
gitarAkustik.ubahHarga(1500000.0);

// Tambahkan stok sebanyak 5 unit
gitarAkustik.tambahStok(5);

// 2. Barang Kedua: Drum Set (Cara ketiga: data lengkap)
// Kode: "DRM-001", Nama: "Drum Set Pearl", Harga: 8.500.000, Stok: 3
BarangMusik drumSet = new BarangMusik("DRM-001", "Drum Set Pearl", 8500000.0, 3);

// Tampilkan informasi kedua barang
System.out.println("=== INVENTARIS TOKO NADAKITA ===");
gitarAkustik.tampilInfo();
drumSet.tampilInfo();
}
}

```

III. Hasil Output Program

```

=== INVENTARIS TOKO NADAKITA ===
Alat musik Gitar Akustik Yamaha | Kode: GTR-001 | Harga: Rp 1500000.0 | Stok: 5 unit
Alat musik Drum Set Pearl | Kode: DRM-001 | Harga: Rp 8500000.0 | Stok: 3 unit

Process finished with exit code 0

```

Penjelasan Hasil:

1. Gitar Akustik:

- Awalnya dibuat dengan harga 0 dan stok 0.
- Setelah ubahHarga, harga menjadi 1.500.000.
- Setelah tambahStok, stok menjadi 5.
- Output mencerminkan data terakhir tersebut.

2. Drum Set:

- Dibuat langsung dengan data lengkap, sehingga output langsung sesuai dengan data saat inisialisasi (Harga 8.500.000, Stok 3).