

**LAPORAN UJIAN TENGAH SEMESTER  
PEMROGRAMAN BERORIENTASI OBJECT**

**PERIODE: SEMESTER GENAP 2024/2025**



**NAMA KELOMPOK:**

**ZENEO AIMAAN – 24104410025  
ENSTIKA ELINDASARI – 24104410006  
NESSA NOVIANTI – 24104410007  
LULU ANGELITA SUGIARTO – 24104410021  
ARIENTA AMANDA PUTRI – 24104410022**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ISLAM BALITAR  
2024/2025**

1. **Pandu sedang mempelajari bangun ruang. Dia mempelajari volume dan luas dari bangun-bangun ruang. Buatlah program java untuk membantu dia menghitung volume dan luas dari kubus, balok, silinder dan bola dengan memperhatikan pseudocode dibawah ini.**

**//KUBUS**

**BANGUNRUANG(int rusuk) [....3 lines]**

**//BALOK**

**BANGUNRUANG(double panjang, double lebar, double tinggi) [....3 lines]**

**//SILINDER**

**BANGUNRUANG(double jari, double tinggi) [....3 lines]**

**Syarat:**

**Gunakan konsep overloading pada constructor dan ambil inputan dari keyboard untuk menghitung panjang, lebar, tinggi, dan jari-jari.**

**Penyelesaian:**

```
import java.util.Scanner;
```

```
public class BangunRuang {
    private double Luas;
    private double Volume;
    private String Nama;

    // Constructor kubus
    public BangunRuang(double rusuk) {
        Nama = "Kubus";
        Luas = 6 * Math.pow(rusuk, 2);
        Volume = Math.pow(rusuk, 3);
    }

    // Constructor balok
    public BangunRuang(double panjang, double lebar, double tinggi) {
        Nama = "Balok";
        Luas = 2 * (panjang * lebar + panjang * tinggi + lebar * tinggi);
        Volume = panjang * lebar * tinggi;
    }

    // Constructor Syylinder
    public BangunRuang(double radius, double tinggi, boolean syylinder) {
        Nama = "Syylinder";
        Luas = 2 * Math.PI * radius * (radius + tinggi);
        Volume = Math.PI * Math.pow(radius, 2) * tinggi;
    }

    // Constructor Bola
    public BangunRuang(double radius, boolean bola, boolean bukansyylinder ) {
        Nama = "bola";
        Luas = 4 * Math.PI * Math.pow(radius, 2);
        Volume = (4.0 / 3.0) * Math.PI * Math.pow(radius, 3);
    }
}
```

```

// Method Display
public void showHasil() {
    System.out.printf("Shape: %s\n", Nama);
    System.out.printf("Volume: %.2f %n", Volume);
    System.out.printf("Luas: %.2f %n", Luas);
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Menghitung Luas dan Volume Kubus
    System.out.println("Masukkan rusuk kubus");
    double edge = scanner.nextDouble();
    BangunRuang kubus = new BangunRuang(edge);
    kubus.showHasil();

    // Menghitung Luas dan Volume Balok
    System.out.println("\nMasukkan panjang balok:");
    double panjang = scanner.nextDouble();
    System.out.println("Masukkan lebar balok:");
    double lebar = scanner.nextDouble();
    System.out.println("Masukkan tinggi balok:");
    double tinggi = scanner.nextDouble();
    BangunRuang balok = new BangunRuang(panjang, lebar, tinggi);
    balok.showHasil();

    // Menghitung Luas dan Volume Syllinder
    System.out.println("\nMasukkan radius syllinder:");
    double syllinderRadius = scanner.nextDouble();
    System.out.println("Masukkan tinggi syllinder");
    double syllindertinggi = scanner.nextDouble();
    BangunRuang syllinder = new BangunRuang(syllinderRadius,
syllindertinggi, true);
    syllinder.showHasil();

    // Menghitung Luas dan Volume Bola
    System.out.println("\nMasukkan radius bola:");
    double bolaRadius = scanner.nextDouble();
    BangunRuang bola = new BangunRuang(bolaRadius, true, true);
    bola.showHasil();

    scanner.close();
}
}

```

## Penjelasan:

1.

```
import java.util.Scanner;

public class BangunRuang {
    private double Luas;
    private double Volume;
    private String Nama;
```

- Import java.util,Scanner = digunakan untuk mengimpor kelas scanner agar dapat menginput dari keyboard bukan langsung dari codingannya
- Public class BangunRuang = digunakan untuk membuat class
- Private double Luas = digunakan untuk menyimpan luas dari bangun ruang yang hanya diakses di class
- Private double Volume = digunakan untuk menyimpan volume dari bangun ruang yang hanya diakses di class
- Private String Nama = digunakan untuk menyimpan variable nama dari bangun ruang yang hanya diakses di class.

2.

```
// Constructor kubus
public BangunRuang(double edge) {
    Nama = "Kubus";
    Luas = 6 * Math.pow(edge, 2);
    Volume = Math.pow(edge, 3);
}
```

- Public BangunRuang(double edge) = merupakan deklarasi constructor untuk kelas tersebut
- Nama = digunakan untuk mengatur nama bentuk menjadi Kubus
- Luas = digunakan untuk menghitung luas kubus dengan rumus tersebut
- Volume = digunakan untuk menghitung volume dengan rumus tersebut

3.

```
// Constructor balok
public BangunRuang(double panjang, double lebar, double tinggi) {
    Nama = "Balok";
    Luas = 2 * (panjang * lebar + panjang * tinggi + lebar * tinggi);
    Volume = panjang * lebar * tinggi;
}
```

- Public BangunRuang(double panjang, double lebar, double tinggi) = merupakan deklarasi constructor untuk kelas tersebut
- Nama = digunakan untuk mengatur nama bentuk menjadi Kubus
- Luas = digunakan untuk menghitung luas sylinder dengan rumus tersebut
- Volume = digunakan untuk menghitung volume dengan rumus tersebut

4.

```
// Constructor Syylinder
public BangunRuang(double radius, double tinggi, boolean syylinder) {
    Nama = "Syylinder";
    Luas = 2 * Math.PI * radius * (radius + tinggi);
    Volume = Math.PI * Math.pow(radius, 2) * tinggi;
}
```

- Public BangunRuang(double panjang, double lebar, double tinggi) = merupakan deklarasi constructor untuk kelas tersebut
- Nama = digunakan untuk mengatur nama bentuk menjadi Kubus
- Luas = digunakan untuk menghitung luas syylinder dengan rumus tersebut
- Volume = digunakan untuk menghitung volume dengan rumus tersebut

5.

```
// Constructor Bola
public BangunRuang(double radius, boolean bola, boolean bukansyylinder ) {
    Nama = "bola";
    Luas = 4 * Math.PI * Math.pow(radius, 2);
    Volume = (4.0 / 3.0) * Math.PI * Math.pow(radius, 3);
}
```

- Public BangunRuang(double radius, boolean bola, boolean bukansyylinder ) = merupakan deklarasi constructor untuk kelas tersebut
- Nama = digunakan untuk mengatur nama bentuk menjadi Kubus
- Luas = digunakan untuk menghitung luas bola dengan rumus tersebut
- Volume = digunakan untuk menghitung volume dengan rumus tersebut

6.

```
// Method Display
public void showHasil() {
    System.out.printf("Shape: %s%n", Nama);
    System.out.printf("Volume: %.2f %n", Volume);
    System.out.printf("Luas: %.2f %n", Luas);
}
```

- Public void showHasil()** = digunakan untuk deklarasi metode yang digunakan untuk menampilkan hasil
- System.out.printf("Shape: %s%n", Nama) = System.out.printf** ini digunakan menampilkan output, serta **Shape:** disini sebagai petunjuk bentuk mana yang akan di keluarkan, dan **%s** merupakan placeholder untuk string (teks) diganti dengan nilai nama yang merupakan nama bentuk (misalnya, "Kubus" atau "Bola"), **%n** digunakan untuk spasi, untuk **nama** : variabel yang menampung nama bentuk, ditetapkan oleh konstruktor saat bentuk dibuat.
- System.out.printf("Volume: %.2f %n", Volume) = System.out.printf** ini digunakan menampilkan output, serta **Volume:** mencetak kata volume, dan **%.2f** merupakan tempat penampung untuk angka dengan desimal, **%n** digunakan untuk spasi, untuk **volume** variabel yang menampung volume bentuk, dihitung oleh konstruktor
- System.out.printf("Luas: %.2f %n", Luas) = System.out.printf** ini digunakan menampilkan output, serta **Luas:** mencetak kata luas, dan **%.2f** merupakan tempat penampung untuk angka dengan desimal, **%n** digunakan untuk spasi, untuk luas variabel yang menampung luas bentuk, dihitung oleh konstruktor

7.

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
  
    // Menghitung Luas dan Volume Kubus  
    System.out.println("Masukkan rusuk kubus");  
    double edge = scanner.nextDouble();  
    BangunRuang kubus = new BangunRuang(edge);  
    kubus.showHasil();  
}
```

- a. Membuat objek Scanner untuk membaca input dari keyboard
- b. **System.out.println** digunakan untuk Menampilkan pesan ke user agar memasukkan nilai rusuk kubus
- c. **double edge = scanner.nextDouble();** Menerima input dari user (angka desimal atau double) dan disimpan ke variabel edge.
- d. Membuat objek BangunRuang untuk **kubus**, dan mengirimkan nilai edge sebagai argumen ke konstruktor
- e. Memanggil metode showHasil() dari objek kubus

```
// Menghitung Luas dan Volume Balok  
System.out.println("\nMasukkan panjang balok:");  
double panjang = scanner.nextDouble();  
System.out.println("Masukkan lebar balok:");  
double lebar = scanner.nextDouble();  
System.out.println("Masukkan tinggi balok:");  
double tinggi = scanner.nextDouble();  
BangunRuang balok = new BangunRuang(panjang, lebar, tinggi);  
balok.showHasil();
```

8.

- a. **scanner.nextDouble();** Ini membaca input dari user berupa angka desimal
- b. **BangunRuang balok = new BangunRuang(panjang, lebar, tinggi);** mengirim data dari input user ke dalam class supaya bisa diolah.

9.

```
// Menghitung Luas dan Volume Syylinder  
System.out.println("\nMasukkan radius syylinder:");  
double syylinderRadius = scanner.nextDouble();  
System.out.println("Masukkan tinggi syylinder");  
double syindertinggi = scanner.nextDouble();  
BangunRuang syylinder = new BangunRuang(syylinderRadius, syindertinggi, true);  
syylinder.showHasil();
```

- a. **BangunRuang silinder = new BangunRuang(cylinderRadius, cylinderTinggi, true);**
- b. Objek silinder dibuat dari kelas BangunRuang dengan memanggil konstruktor yang menerima tiga parameter

10.

```
// Menghitung Luas dan Volume Bola  
System.out.println("\nMasukkan radius bola:");  
double bolaRadius = scanner.nextDouble();  
BangunRuang bola = new BangunRuang(bolaRadius, true, true);  
bola.showHasil();  
  
scanner.close();  
}
```

- a. **scanner.close();** Metode close() dipanggil untuk menutup objek scanner, sehingga sumber daya yang digunakan untuk membaca input dari pengguna dibebaskan.

## Output:

```
run:
Masukkan rusuk kubus
5
Shape: Kubus
Volume: 125,00
Luas: 150,00

Masukkan panjang balok:
7
Masukkan lebar balok:
9
Masukkan tinggi balok:
8
Shape: Balok
Volume: 504,00
Luas: 382,00

Masukkan radius sylinder:
2
Masukan tinggi sylinder
3
Shape: Sylinder
Volume: 37,70
Luas: 62,83

Masukkan radius bola:
7
Shape: bola
Volume: 1436,76
Luas: 615,75
BUILD SUCCESSFUL (total time: 43 seconds)
```

2. Seseorang bernama Jono ingin mendirikan sebuah toko smartphone dengan berbagai merk dari vendor. Dikarenakan regulasi di Indonesia, mengharuskan barang-barang yang di impor dari luar negeri diberikan pajak tertentu oleh Bea Cukai. Jono meminta kalian untuk membuat program berbasis OOP menggunakan bahasa Java untuk membantu menghitung harga smartphone yang sudah include dengan PPN. List vendor, tipe dan harga dari smartphone yang Jono beli sebagai berikut:

Vendor	Tipe	Harga
Samsung	A54	Rp 5,500,000.00
Oppo	Reno 13	Rp 8,800,000.00
Xiaomi	Poco X6 Pro	Rp 4,400,000.00

**PPN:**

- Jika harga > Rp 4,000,000,00 maka PPN 1%
- Jika harga > Rp 4,500,000,00 sampai Rp 8,000,000,00 maka PPN 2%
- Jika harga > Rp 8,000,000,00 maka PPN 5%

**Syarat:**

- a) Buatlah 1 class Pajak berisi enkapsulasi dari variable vendor, tipe, harga.
- b) Buatlah 1 class Main berisi main method serta pemanggilan objek dari class Pajak untuk menghitung harga PPN.
- c) Untuk nama vendor, tipe, harga wajib menggunakan inputan dari keyboard.

**Penyelesaian:**

a) **Class Pajak**

```
public class Pajak {
    // Variabel dengan enkapsulasi (private)
    private String vendor;
    private String tipe;
    private double harga;

    // Constructor
    public Pajak(String vendor, String tipe, double harga) {
        this.vendor = vendor;
        this.tipe = tipe;
        this.harga = harga;
    }

    // Getter dan Setter
    public String getVendor() {
        return vendor;
    }

    public void setVendor(String vendor) {
        this.vendor = vendor;
    }

    public String getTipe() {
        return tipe;
    }
}
```



```

    public void setType(String tipe) {
        this.tipe = tipe;
    }

    public double getHarga() {
        return harga;
    }

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

    // Menghitung PPN berdasarkan aturan
    public double hitungPPN() {
        if (harga > 8000000) {
            return harga * 0.05; // PPN 5%
        } else if (harga > 4500000) {
            return harga * 0.02; // PPN 2%
        } else if (harga > 4000000) {
            return harga * 0.01; // PPN 1%
        }
        return 0; // Tidak ada PPN jika harga <= Rp 4.000.000
    }

    // Menghitung total harga (harga + PPN)
    public double hitungTotalHarga() {
        return harga + hitungPPN();
    }
}

```

### Penjelasan:

1.

```

//
public class Pajak {
    // Variabel dengan enkapsulasi (private)
    private String vendor;
    private String tipe;
    private double harga;
}

```

- public class Pajak : digunakan untuk membuat class
- private String vendor : menyimpan nama vendor
- private String tipe : menyimpan type smartphone
- private double harga : menyimpan harga smartphone

2.

```

public Pajak(String vendor, String tipe, double harga) {
    this.vendor = vendor;
    this.tipe = tipe;
    this.harga = harga;
}

```

- Public Pajak (String vendor, String tipe, double harga) : menginisialisasi nilai awal variable vendor, tipe, dan harga saat objek dibuat.
- This.vendor = vendor; : baris ini menugaskan nilai parameter vendor ke variable instance vendor, memastikan data yang dimaksudkan disimpan.

- This.tipe = tipe; : menugaskan nilai parameter tipe ke variable instance tipe.
- This.harga = harga; : menugaskan nilai parameter harga ke variable instance harga.

3.

```
// Getter dan Setter
public String getVendor() {
    return vendor;
}

public void setVendor(String vendor) {
    this.vendor = vendor;
}

public String getType() {
    return tipe;
}

public void setType(String tipe) {
    this.tipe = tipe;
}

public double getHarga() {
    return harga;
}

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

- Public String getVendor() { :
- Mendefinisikan method getVendor bertipe public yang mengembalikan nilai String.
- Method ini adalah getter, digunakan untuk mengambil nilai vendor dari luar kelas sesuai prinsip enkapsulasi.
- return vendor : Mengembalikan nilai variable vender, karena vendor bersifat private, ini adalah satu-satunya cara legal untuk mengaksesnya dari luar.
- } (penutup getVendor) : Menutup blok method getVendor
- public String getType() { : Getter untuk variable tipe, serupa dengan getVendor.
- return tipe; : Mengembalikan nilai tipe.
- } (penutup getType) : Menutup blok method getType.
- public void setType(String tipe) { : Setter untuk variabel tipe.
- this.tipe = tipe; : Menugaskan nilai parameter tipe ke variabel instance tipe.
- } (penutup setType) : Menutup blok method setType.
- public double getHarga() { : Getter untuk variabel harga.
- return harga; : Mengembalikan nilai harga.
- } (penutup getHarga) : Menutup blok method getHarga.
- public void setHarga(double harga) { : Setter untuk variabel harga.
- this.harga = harga; : Menugaskan nilai parameter harga ke variabel instance harga.
- } (penutup setHarga) : Menutup blok method setHarga.

4.

```
// Menghitung PPN berdasarkan aturan
public double hitungPPN() {
    if (harga > 8000000) {
        return harga * 0.05; // PPN 5%
    } else if (harga > 4500000) {
        return harga * 0.02; // PPN 2%
    } else if (harga > 4000000) {
        return harga * 0.01; // PPN 1%
    }
    return 0; // Tidak ada PPN jika harga <= Rp 4.000.000
}
```

- a. public double hitungPPN() { : Mendefinisikan method hitungPPN bertipe public yang mengembalikan nilai double. Method ini akan menghitung PPN berdasarkan aturan yang diberikan (1%, 2%, atau 5%).
- b. if (harga > 8000000) { : Kondisi pertama, Jika harga lebih dari Rp 8,000,000, maka PPN 5%. Operator > membandingkan nilai harga dengan 8,000,000.
- c. return harga \* 0.05; // PPN 5% : Jika kondisi terpenuhi, menghitung PPN dengan mengalikan harga dengan 0.05 (5%). return mengakhiri method dan mengembalikan hasil perhitungan. Komentar // PPN 5% untuk dokumentasi.
- d. else if (harga > 4500000) { : Kondisi kedua: Jika harga lebih dari Rp 4,500,000 (dan ≤ Rp 8,000,000 karena kondisi sebelumnya gagal), maka PPN 2%. else if memastikan hanya satu kondisi yang dieksekusi.
- e. return harga \* 0.02; // PPN 2% : Menghitung PPN 2% dari harga dan mengembalikannya.
- f. else if (harga > 4000000) { : Kondisi ketiga: Jika harga lebih dari Rp 4,000,000 (dan ≤ Rp 4,500,000), maka PPN 1%.
- g. return harga \* 0.01; // PPN 1% : Menghitung PPN 1% dari harga dan mengembalikannya.
- h. return 0; // Tidak ada PPN jika harga <= Rp 4.000.000 : Jika semua kondisi gagal (artinya harga ≤ Rp 4,000,000), maka PPN = 0. Komentar menjelaskan alasan pengembalian 0.
- i. } (penutup hitungPPN) Menutup blok method hitungPPN.

5.

```
// Menghitung total harga (harga + PPN)
public double hitungTotalHarga() {
    return harga + hitungPPN();
}
}
```

- a. public double hitungTotalHarga() { : Mendefinisikan method hitungTotalHarga bertipe public yang mengembalikan nilai double. Method ini menghitung total harga dengan menambahkan harga dan PPN.
- b. return harga + hitungPPN(); Menjumlahkan nilai harga dengan hasil dari hitungPPN(). hitungPPN() dipanggil untuk mendapatkan nilai PPN secara dinamis.
- c. } (penutup hitungTotalHarga) : Menutup blok method hitungTotalHarga.
- d. } (penutup class Pajak) : Menutup seluruh definisi kelas Pajak.

## b) Class Main

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input dari pengguna
        System.out.println("Masukkan nama vendor:");
        String vendor = scanner.nextLine();

        System.out.println("Masukkan tipe smartphone:");
        String tipe = scanner.nextLine();

        System.out.println("Masukkan harga smartphone (Rp):");
        double harga = scanner.nextDouble();

        // Membuat objek Pajak
        Pajak smartphone = new Pajak(vendor, tipe, harga);

        // Menghitung PPN dan total harga
        double ppn = smartphone.hitungPPN();
        double totalHarga = smartphone.hitungTotalHarga();

        // Menampilkan hasil
        System.out.println("\nDetail Smartphone:");
        System.out.println("Vendor: " + smartphone.getVendor());
        System.out.println("Tipe: " + smartphone.getTipe());
        System.out.printf("Harga: Rp %.2f\n", smartphone.getHarga());
        System.out.printf("PPN: Rp %.2f\n", ppn);
        System.out.printf("Total Harga (termasuk PPN): Rp %.2f\n",
totalHarga);

        scanner.close();
    }
}
```

### Penjelasan:

1. Import java.util.Scanner;
  - Digunakan untuk mengimpor Scanner dari paket java.util.
  - Scanner diperlukan untuk membaca input dari keyboard, sesuai syarat (c).
2. Public class main {
  - Digunakan untuk mendeklarasikan kelas main yang bersifat public.
  - Kelas ini akan berisi method main sebagai titik masuk program.
3. public static void main(String[] args) {
  - Mendefinisikan method main yang bersifat public dan static.
  - public memungkinkan method diakses dari luar, static memungkinkan pemanggilan tanpa instansiasi objek, dan void menunjukkan method tidak mengembalikan nilai.
  - String[] args adalah parameter untuk menerima argumen baris perintah (tidak digunakan di sini).
  - Method ini memenuhi syarat (b) untuk class Main dengan main method.
4. Scanner scanner = new Scanner(System.in);

- Membuat objek scanner dari kelas Scanner untuk membaca input dari System.in (keyboard).
  - Scanner akan digunakan untuk mengambil input vendor, tipe, dan harga.
5. `System.out.println("Masukkan nama vendor:");`
    - Mencetak pesan ke konsol untuk meminta pengguna memasukkan nama vendor.
    - `\n` menambahkan baris baru setelah pesan.
  6. `String vendor = scanner.nextLine();`
    - Membaca input pengguna sebagai String dan menyimpannya ke variabel vendor.
    - `nextLine()` mengambil seluruh baris teks hingga pengguna menekan Enter, sesuai syarat (c).
  7. `System.out.println("Masukkan tipe smartphone:");`
    - Mencetak pesan untuk meminta input tipe smartphone.
  8. `String tipe = scanner.nextLine();`
    - Membaca input tipe sebagai String dan menyimpannya ke variabel tipe.
  9. `System.out.println("Masukkan harga smartphone (Rp):");`
    - Mencetak pesan untuk meminta input harga.
  10. `double harga = scanner.nextDouble();`
    - Membaca input harga sebagai double dan menyimpannya ke variabel harga.
    - `nextDouble()` mengambil angka desimal, sesuai syarat (c) untuk input harga.
  11. `Pajak smartphone = new Pajak(vendor, tipe, harga);`
    - Membuat objek smartphone dari kelas Pajak dengan memanggil constructor Pajak.
    - Parameter vendor, tipe, dan harga dari input pengguna dilewatkan ke constructor.
    - Ini memenuhi syarat (b) untuk pemanggilan objek dari class Pajak.
  12. `double ppn = smartphone.hitungPPN();`
    - Memanggil method `hitungPPN()` dari objek `smartphone` untuk menghitung PPN.
    - Hasilnya disimpan ke variabel `ppn` bertipe double.
  13. `double totalHarga = smartphone.hitungTotalHarga();`
    - Memanggil method `hitungTotalHarga()` dari objek `smartphone` untuk menghitung total harga.
    - Hasilnya disimpan ke variabel `totalHarga`.
  14. `System.out.println("\nDetail Smartphone:");`
    - Mencetak judul "Detail Smartphone:" dengan baris baru sebelumnya.
  15. `System.out.println("Vendor: " + smartphone.getVendor());`
    - Mencetak nilai vendor dengan memanggil `getVendor()` dari objek `smartphone`.
    - Operator `+` menggabungkan string "Vendor: " dengan nilai yang dikembalikan.
  16. `System.out.println("Tipe: " + smartphone.getTipe());`
    - Mencetak nilai tipe dengan memanggil `getTipe()`.
  17. `System.out.printf("Harga: Rp %.2f%n", smartphone.getHarga());`
    - Mencetak nilai harga dengan format khusus menggunakan `printf`.
    - `%.2f` memastikan angka desimal ditampilkan dengan 2 digit (misalnya 5500000.00).
    - `%n` menambahkan baris baru.
    - `smartphone.getHarga()` mengambil nilai harga.
  18. `System.out.printf("PPN: Rp %.2f%n", ppn);`
    - Mencetak nilai `ppn` dengan format yang sama.
  19. `System.out.printf("Total Harga (termasuk PPN): Rp %.2f%n", totalHarga);`
    - Mencetak nilai `totalHarga` dengan format yang sama.
  20. `scanner.close();`
    - Menutup objek scanner untuk melepaskan sumber daya input.

- Ini adalah praktik baik untuk mengelola sumber daya.
21.       } (penutup main)
- Menutup blok method main.
22.       } (penutup class Main)
- Menutup seluruh definisi kelas Main.

### c) Output

- PPN 1%

---

```
run:
Masukkan nama vendor:
samsung
Masukkan tipe smartphone:
A54
Masukkan harga smartphone (Rp):
4000000

Detail Smartphone:
Vendor: samsung
Tipe: A54
Harga: Rp 4000000,00
PPN: Rp 0,00
Total Harga (termasuk PPN): Rp 4000000,00
BUILD SUCCESSFUL (total time: 1 minute 7 seconds)
```

- PPN 2%

---

```
run:
Masukkan nama vendor:
oppo
Masukkan tipe smartphone:
reno 13
Masukkan harga smartphone (Rp):
8800000

Detail Smartphone:
Vendor: oppo
Tipe: reno 13
Harga: Rp 8800000,00
PPN: Rp 440000,00
Total Harga (termasuk PPN): Rp 9240000,00
BUILD SUCCESSFUL (total time: 55 seconds)
```

- PPN 5%

```
run:
Masukkan nama vendor:
oppo
Masukkan tipe smartphone:
reno 13
Masukkan harga smartphone (Rp):
8800000

Detail Smartphone:
Vendor: oppo
Tipe: reno 13
Harga: Rp 8800000,00
PPN: Rp 440000,00
Total Harga (termasuk PPN): Rp 9240000,00
BUILD SUCCESSFUL (total time: 32 seconds)
```