

Nama: Harry LBI  
NIM: A11.2023.15332

## LATIHAN 1

```
package pratikum4.latihan1;

public class Penjualan {

    private String kode;
    private String nama;
    private float harga;
    private int jumlah;

    public void setData(String kode, String nama, float harga, int
jumlah) {
        this.kode = kode;
        this.nama = nama;
        this.harga = harga;
        this.jumlah = jumlah;
    }

    public float getTotalPembelian() {
        return harga * jumlah;
    }
}
```

Atribut ini bersifat private, artinya hanya bisa diakses dalam kelas Penjualan:

kode → Menyimpan kode barang.

nama → Menyimpan nama barang.

harga → Menyimpan harga satuan barang.

jumlah → Menyimpan jumlah barang yang dibeli.

### Metode setData

Metode ini digunakan untuk mengatur kode barang, nama barang, harga, dan jumlah barang.

this.kode = kode; memastikan bahwa variabel (kode) mendapatkan nilai dari parameter (kode).

Data yang diinput dengan setData() akan digunakan oleh metode getTotalPembelian(), getBonus(), dan cetakNota().

### Metode getTotalPembelian

Menghitung total pembelian dengan mengalikan harga dengan jumlah

```
21     public String getBonus() {
22         float total = getTotalPembelian();
23         if (total >= 5000000 && jumlah > 5) {
24             return "Dapat bonus setrika";
25         } else if (total >= 100000 && jumlah > 3) {
26             return "Dapat bonus payung";
27         } else if (total >= 50000 && jumlah > 2) {
28             return "Dapat bonus ballpoint";
29         } else {
30             return "Tidak dapat bonus";
31         }
32     }
33
34     // Perbaikan: Menambahkan return type void
35     public void cetakNota() {
36         System.out.println("==== NOTA PEMBELIAN =====");
37         System.out.println("Kode Barang      : " + kode);
38         System.out.println("Nama Barang      : " + nama);
39         System.out.println("Harga Satuan     : Rp " + harga);
40         System.out.println("Jumlah           : " + jumlah);
41         System.out.println("Total Harga      : Rp " + getTotalPembelian());
42         System.out.println("Bonus            : " + getBonus());
43         System.out.println("=====");
44     }
45 }
```

### Metode getBonus

Metode ini menggunakan hasil dari getTotalPembelian() untuk menentukan apakah pembelian berhak mendapatkan bonus tertentu.

### Metode cetakNota

untuk Menampilkan semua informasi pembelian, termasuk kode, nama, harga satuan, jumlah barang, total harga, dan bonus.

```
Run | Debug
public static void main(String[] args) {
    Penjualan penjualan1 = new Penjualan();
    Scanner input = new Scanner(System.in);
    boolean lanjut = true;

    while (lanjut) {
        System.out.print("Masukkan kode barang:");
        String kode = input.next();

        input.nextLine(); // Membersihkan buffer
        System.out.print("Masukkan nama barang:");
        String nama = input.nextLine();

        System.out.print("Masukkan harga barang:");
        int harga = input.nextInt();

        System.out.print("Masukkan jumlah barang:");
        int jumlah = input.nextInt();
        input.nextLine(); // Membersihkan buffer

        penjualan1.setData(kode, nama, harga, jumlah);
        penjualan1.cetakNota();

        System.out.print("Apakah ingin memasukkan data lagi? (y/n)");
        String pilihan = input.nextLine();

        if (pilihan.equalsIgnoreCase("n")) {
            lanjut = false;
        }
    }
}
```

Penjualan penjualan1 = new Penjualan(); → Membuat objek penjualan1 dari kelas Penjualan.

Scanner input = new Scanner(System.in); → Scanner untuk menerima input pengguna.

boolean lanjut = true; → Variabel kontrol untuk loop while, agar pengguna bisa menginput lebih dari satu transaksi.

while (lanjut) {

Loop **while** memungkinkan pengguna untuk **memasukkan lebih dari satu transaksi**.

**Akan terus berjalan** sampai pengguna memilih "n" pada pertanyaan akhir.

input.next(); digunakan untuk membaca kode barang.

Pengguna harus mengetikkan kode barang dan menekan Enter.

input.nextLine(); pertama membersihkan buffer scanner agar tidak terjadi bug input loncat setelah input.next().

input.nextLine(); kedua membaca nama barang.

input.nextInt(); membaca harga barang (integer).

input.nextInt(); membaca jumlah barang.

input.nextLine(); membersihkan buffer setelah nextInt() agar tidak terjadi kesalahan input saat membaca string.

penjualan1.setData(kode, nama, harga, jumlah);

Menyimpan data barang ke dalam objek penjualan1.

penjualan1.cetakNota();

Menampilkan nota pembelian, yang mencetak:

Kode Barang, Nama Barang, Harga Satuan, Jumlah, Total Harga, dan Bonus

## Latiahn 2

```
// Hitung akar dengan menerima nilai a, b, dan c
public Hitung(double a, double b, double c) {
    this.a = a;
    this.b = b;
    this.c = c;
}

// Metode untuk menghitung diskriminan
public double getDeterminan() {
    return (b * b) - (4 * a * c);
}
```

Konstruktor ini digunakan untuk menginisialisasi nilai a, b, dan c saat objek Hitung dibuat.

Dengan menggunakan konstruktor ini, pengguna dapat langsung memberikan nilai untuk koefisien a, b, dan c saat membuat objek.

**Metode getDeterminan()** menghitung **diskriminan** (D) dari persamaan kuadrat, yang digunakan untuk menentukan jenis akar persamaan tersebut.

```
// Metode untuk menghitung akar persamaan kuadrat
public void hitungAkar() {
    double D = getDeterminan();

    if (D > 0) {
        double x1 = (-b + Math.sqrt(D)) / (2 * a);
        double x2 = (-b - Math.sqrt(D)) / (2 * a);
        System.out.println("Akar-akar real: x1 = " + x1 + ", x2 = " + x2);
    } else if (D == 0) {
        double x = -b / (2 * a);
        System.out.println("Akar kembar: x1 = x2 = " + x);
    } else {
        double realPart = -b / (2 * a);
        double imagPart = Math.sqrt(-D) / (2 * a);
        System.out.println("Akar imajiner: x1 = " + realPart + " + " + imagPart + "i");
        System.out.println("                x2 = " + realPart + " - " + imagPart + "i");
    }
}
}
```

**Metode hitungAkar()** menghitung dan menampilkan **akar persamaan kuadrat** berdasarkan nilai diskriminan (D).

**Jika  $D > 0$  (Akar Real Berbeda):**

**Jika  $D == 0$  (Akar Kembar):**

**Jika  $D < 0$  (Akar Imajiner)**

```
1  package pratikum4.latihan2;
2
3  import java.util.Scanner;
4
5  public class Main {
6      Run | Debug
7      public static void main(String[] args) {
8          Scanner scanner = new Scanner(System.in);
9          char ulangi;
10
11          do {
12              System.out.print("Masukkan nilai a: ");
13              double a = scanner.nextDouble();
14              System.out.print("Masukkan nilai b: ");
15              double b = scanner.nextDouble();
16              System.out.print("Masukkan nilai c: ");
17              double c = scanner.nextDouble();
18
19              Hitung persamaan = new Hitung(a, b, c);
20              persamaan.hitungAkar();
21
22              System.out.print("Input data lagi? [Y/N]: ");
23              ulangi = scanner.next().charAt(0);
24          } while (ulangan == 'Y' || ulangi == 'y');
25
26          scanner.close();
27      }
28  }
```

```
Scanner scanner = new Scanner(System.in);
```

Membuat objek **Scanner** untuk menerima input dari pengguna.

```
char ulangi;
```

Variabel **ulangi** digunakan untuk menentukan apakah pengguna ingin mengulang input atau tidak.

disini menggunakan perulangan do while untuk menginput dan menghitung akar

di bagian do pengguna akan disuruh input

dan di while inputan itu akan diproses dengan

**Membuat objek Hitung** dengan nilai a, b, dan c.

**Memanggil hitungAkar()** untuk menghitung akar-akar dari persamaan kuadrat berdasarkan nilai **diskriminan**.

**Program akan menanyakan apakah pengguna ingin menginput ulang data.**

Jika **pengguna mengetik 'Y' atau 'y'**, program akan mengulang.

Jika **pengguna mengetik 'N' atau 'n'**, program akan berhenti.

# LATIHAN 3

```
package pratikum4.latihan3;

public class Konversi {
    private int totalDetik;

    public Konversi(int totalDetik) {
        this.totalDetik = totalDetik;
    }

    public void hitungKonversi() {
        int hari = totalDetik / 86400;
        int sisaDetik = totalDetik % 86400;

        int jam = sisaDetik / 3600;
        sisaDetik %= 3600;

        int menit = sisaDetik / 60;
        int detik = sisaDetik % 60;

        System.out.println("Detik      : " + totalDetik);
        System.out.println("Hari       : " + hari);
        System.out.println("Jam        : " + jam);
        System.out.println("Menit      : " + menit);
        System.out.println("Detik      : " + detik);
    }
}
```

Konstruktor digunakan untuk **menginisialisasi** nilai total detik saat objek **Konversi** dibuat.

## metode hitungKonversi()

metode ini berguna untuk mengkonkonversu detik ke menit, jam dan hari yang akan diinput nanti

```
1  package pratikum4.latihan3;
2
3  import java.util.Scanner;
4
5  public class Main {
6      Run | Debug
7      public static void main(String[] args) {
8          Scanner scanner = new Scanner(System.in);
9          char ulangi;
10
11          do {
12              System.out.print("Masukkan jumlah detik: ");
13              int detik = scanner.nextInt();
14
15              Konversi konversi = new Konversi(detik);
16              konversi.hitungKonversi();
17
18              System.out.print("Ingin input lagi? [Y/T]: ");
19              ulangi = scanner.next().charAt(0);
20          } while (ulangi == 'Y' || ulangi == 'y');
21
22          scanner.close();
23      }
24  }
```

Disini, **do-while**

digunakan untuk menangkap input dari user secara berulang sampai user memilih untuk berhenti. Pada bagian do, program akan meminta input jumlah detik dari user, kemudian membuat objek Konversi untuk memproses dan menampilkan hasil konversinya.