

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**  
**JOBSHEET PERTEMUAN KE-2**



**RIO TRI PRAYOGO**

**TI 1A**

**26**

**2341720236**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2024**

## Praktikum 1 : Deklarasi Class, Atribut, dan Method

### Percobaan :

```
package minggu2;

public class Buku26 {

    String judul, pengarang;
    int halaman, stok, harga;

    void tampilInformasi() {
        System.out.println("Judul: " + judul);
        System.out.println("Pengarang: " + pengarang);
        System.out.println("Jumlah halaman: " + halaman);
        System.out.println("Sisa stok: " + stok);
        System.out.println("Harga: Rp " + harga);
    }

    void terjual(int jml) {
        stok -= jml;
    }

    void restock(int jml) {
        stok += jml;
    }

    void gantiHarga(int hrg) {
        harga = hrg;
    }

}
```

### Verifikasi :

```
Error: Main method not found in class minggu2.Buku26, please define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
```

### Pertanyaan :

1. Sebutkan dua karakteristik class atau object!  
= Class atau Object memiliki dua karakteristik yaitu **Mempunyai Sesuatu** dan **Melakukan Sesuatu**.

2. Perhatikan class **Buku** pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Buku? Sebutkan apa saja atributnya!  
= Class Buku sendiri memiliki 5 buah atribut yaitu judul dan pengarang yang bertipe data *String*, serta halaman, stok, dan, harga yang bertipe data *int*.
3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!  
= Class Buku tersebut memiliki 4 buah method yaitu:
  - tampilInformasi(): void
  - terjual(jml: int): void
  - restock(n: int): void
  - gantiHarga(hrg: int): int
4. Perhatikan method **terjual()** yang terdapat di dalam class **Buku**. Modifikasi isi method tersebut sehingga proses pengurangan hanya dapat dilakukan jika stok masih ada (lebih besar dari 0)!  
= Pengubahan method tersebut bisa dilakukan dengan menambahkan pemilihan *if/else* seperti code dibawah ini:

```
void terjual(int jml) {  
    if (stok > 0) {  
        stok -= jml;  
    } else {  
        System.out.println("Tidak ada stok");  
    }  
}
```

5. Menurut Anda, mengapa method **restock()** mempunyai satu parameter berupa bilangan int?  
= Menurut saya method **restock()** memiliki satu parameter berupa int karena method **restock()** sendiri merupakan method yang berfungsi hanya untuk menambahkan stok yang ada Dimana variabel stok merupakan variabel yang bertipe *int*.
6. **Commit dan push kode program ke Github**



## Praktikum 2 : Instansiasi Object, serta Mengakses Atribut dan Method

### Percobaan :

```
package minggu2;  
  
public class BukuMain26 {  
    public static void main(String[] args) {  
        Buku26 bk1 = new Buku26();  
        bk1.judul = "Today Ends Tomorrow Comes";  
        bk1.pengarang = "Denanda Pratiwi";  
        bk1.halaman = 198;  
        bk1.stok = 13;  
        bk1.harga = 71000;
```

```

        bk1.tampilInformasi();

        bk1.terjual(5);

        bk1.gantiHarga(6000);

        bk1.tampilInformasi();

    }

}

```

#### Verifikasi :

```

Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 71000
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 8
Harga: Rp 6000

```

#### Pertanyaan :

1. Pada class **BukuMain**, tunjukkan baris kode program yang digunakan untuk proses instansiasi!  
Apa nama object yang dihasilkan?  
= Pada class **BukuMain** instansiasi ditunjukkan pada line:

```
Buku26 bk1 = new Buku26();
```

Yang dimana menghasilkan object yang diberi nama "*bk1*" yang diambil dari class **Buku26**.

2. Bagaimana cara mengakses atribut dan method dari suatu objek?  
= Pengaksesan atribut dan method dari suatu object dapat dilakukan dengan menggunakan syntax:

```

bk1.judul
bk1.tampilInformasi();

```

Dimana "**judul**" dan "**tampilInformasi()**" merupakan atribut dan method dari class yang diambil dari "*bk1*" yang merupakan hasil dari instansiasi.

3. Mengapa hasil output pemanggilan method **tampilInformasi()** pertama dan kedua berbeda?  
= Karena syntax sebelumnya terjadi method "**terjual()**" dan "**gantiHarga()**" yang dimana merubah isi dari atribut *harga* dan *stok*.

#### Praktikum 3 : Membuat Konstruktor

##### Percobaan :

##### Object

```

package minggu2;

public class Buku26 {
    String judul, pengarang;

```

```
int halaman, stok, harga;

void tampilInformasi() {
    System.out.println("Judul: " + judul);
    System.out.println("Pengarang: " + pengarang);
    System.out.println("Jumlah halaman: " + halaman);
    System.out.println("Sisa stok: " + stok);
    System.out.println("Harga: Rp " + harga);
}

void terjual(int jml) {
    if (stok > 0) {
        stok -= jml;
    } else {
        System.out.println("Tidak ada stok");
    }
}

void restock(int jml) {
    stok += jml;
}

void gantiHarga(int hrg) {
    harga = hrg;
}

public Buku26() {

}

public Buku26(String jud, String pg, int hal, int stok, int har) {
    judul = jud;
    pengarang = pg;
    halaman = hal;
    this.stok = stok;
    harga = har;
}
```

```
}  
  
}
```

## Main

```
package minggu2;  
  
public class BukuMain26 {  
    public static void main(String[] args) {  
        Buku26 bk1 = new Buku26();  
        bk1.judul = "Today Ends Tomorrow Comes";  
        bk1.pengarang = "Denanda Pratiwi";  
        bk1.halaman = 198;  
        bk1.stok = 13;  
        bk1.harga = 71000;  
  
        bk1.tampilInformasi();  
        bk1.terjual(5);  
        bk1.gantiHarga(6000);  
        bk1.tampilInformasi();  
  
        Buku26 bk2 = new Buku26("Self Reward", "Maheera Ayesha", 160, 29,  
59000);  
        bk2.terjual(11);  
        bk2.tampilInformasi();  
    }  
}
```

## Verifikasi :

```
Judul: Today Ends Tomorrow Comes  
Pengarang: Denanda Pratiwi  
Jumlah halaman: 198  
Sisa stok: 13  
Harga: Rp 71000  
Judul: Today Ends Tomorrow Comes  
Pengarang: Denanda Pratiwi  
Jumlah halaman: 198  
Sisa stok: 8  
Harga: Rp 6000  
Judul: Self Reward  
Pengarang: Maheera Ayesha  
Jumlah halaman: 160  
Sisa stok: 18  
Harga: Rp 59000
```

### Pertanyaan :

1. Pada class **Buku** di Percobaan 3, tunjukkan baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter!

= Deklarasi konstruktor berparameter dalam class **Buku** terdapat pada line:

```
public Buku26(String jud, String pg, int hal, int stok, int har) {  
    judul = jud;  
    pengarang = pg;  
    halaman = hal;  
    this.stok = stok;  
    harga = har;  
}
```

2. Perhatikan class **BukuMain**. Apa sebenarnya yang dilakukan pada baris program berikut?

```
Buku bk2 = new Buku(jud:"Self Reward", pg:"Maheera Ayesha", hal:160, stok:29, har:59000);
```

= Baris program diatas melakukan instansiasi pada konstruktor berparameter yang diambil dari class **Buku**.

3. Hapus konstruktor default pada class **Buku**, kemudian compile dan run program. Bagaimana hasilnya? Jelaskan mengapa hasilnya demikian!

= Jika konstruktor default pada class **Buku** dihapus maka jika di run akan terjadi error:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:  
The constructor Buku26() is undefined  
  
at minggu2.BukuMain26.main(BukuMain26.java:5)
```

Hal ini dapat terjadi karena jika konstruktor default dihapus maka hanya akan tersisa konstruktor berparameter dimana instansiasi dalam class **BukuMain** juga harus dalam bentuk berparameter karena konstruktor default tidak ada dan tidak bisa dipanggil.

4. Setelah melakukan instansiasi object, apakah method di dalam class **Buku** harus diakses secara berurutan? Jelaskan alasannya!  
= Instansiasi object dari class dalam pengaksesannya tidak harus dilakukan secara berurutan, instansiasi ini dapat dilakukan sesuai dengan kebutuhan asalkan instansiasi tersebut sesuai dengan konstruktor yang ada dalam maka object tersebut bisa di istansiasi meskipun tidak urut.
5. Buat object baru dengan nama **buku<NamaMahasiswa>** menggunakan konstruktor berparameter dari class **Buku**!

```
Buku26 Rio = new Buku26("One Piece", "Eiichiro Oda", 80, 15, 35000);  
Rio.terjual(5);  
Rio.tampilInformasi();
```

Output:

```
Judul: One Piece  
Pengarang: Eiichiro Oda  
Jumlah halaman: 80  
Sisa stok: 15  
Harga: Rp 35000
```

6. Commit dan push kode program ke Github

```
feat: add konstruktor default n parameter  
riotrip committed 16 minutes ago
```

ce75331

### Latihan Praktikum :

1. Pada class Buku yang telah dibuat, tambahkan tiga method yaitu **hitungHargaTotal()**, **hitungDiskon()**, dan **hitungHargaBayar()** dengan penjelasan sebagai berikut:
  - Method **hitungHargaTotal()** digunakan untuk menghitung harga total yang merupakan perkalian antara harga dengan jumlah buku yang terjual
  - Method **hitungDiskon()** digunakan untuk menghitung diskon dengan aturan berikut:
    - Jika harga total lebih dari 150000, maka harga didiskon sebesar 12%
    - Jika harga total antara 75000 sampai 150000, maka harga didiskon sebesar 5%
    - Jika harga total kurang dari 75000, maka harga tidak didiskon
  - Method **hitungHargaBayar()** digunakan untuk menghitung harga total setelah dikurangi diskon

### Object :

```
package minggu2;

public class Buku26 {
    String judul, pengarang;
    int halaman, stok, harga, totHrg;
    double hargaBayar, diskon = 0.0;

    void tampilInformasi() {
        System.out.println("Judul: " + judul);
        System.out.println("Pengarang: " + pengarang);
        System.out.println("Jumlah halaman: " + halaman);
        System.out.println("Sisa stok: " + stok);
        System.out.println("Harga: Rp " + harga);
        System.out.println("Total harga penjualan Rp " + totHrg);
        System.out.println("Harga bayar setelah diskon: " + hargaBayar);
        System.out.println("=====");
    }

    void terjual(int jml) {
        if (stok > 0) {
            stok -= jml;
            System.out.println("Terjual: " + jml + " buku");
            hitungTotalHarga(jml);
        } else {
            System.out.println("Tidak ada stok");
        }
    }
}
```



```

    }

    void restock(int jml) {
        stok += jml;
    }

    void gantiHarga(int hrg) {
        harga = hrg;
    }

    void hitungTotalHarga(int jml) {
        totHrg = harga * jml;
    }

    void hitungDiskon() {
        if (totHrg > 150000) {
            diskon = 0.12 * totHrg;
            System.out.println("Anda mendapatkan diskon sebesar 12%");
        } else if (totHrg >= 75000 && totHrg <= 150000) {
            diskon = 0.05 * totHrg;
            System.out.println("Anda mendapatkan diskon sebesar 5%");
        } else {
            System.out.println("Anda tidak mendapatkan diskon");
        }
        hargaBayar(diskon);
    }

    void hargaBayar(double disc) {
        hargaBayar = totHrg - disc;
    }

    public Buku26() {

    }

    public Buku26(String jud, String pg, int hal, int stok, int har) {

```

```
        judul = jud;  
        pengarang = pg;  
        halaman = hal;  
        this.stok = stok;  
        harga = har;  
    }  
}
```

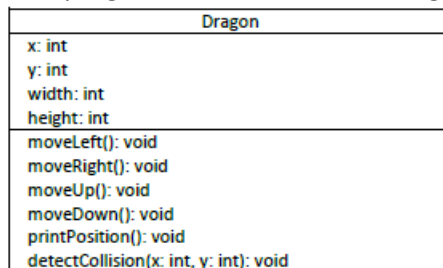
### Main :

```
package minggu2;  
  
public class BukuMain26 {  
    public static void main(String[] args) {  
        Buku26 bk1 = new Buku26();  
        bk1.judul = "Today Ends Tomorrow Comes";  
        bk1.pengarang = "Denanda Pratiwi";  
        bk1.halaman = 198;  
        bk1.stok = 13;  
        bk1.harga = 71000;  
  
        bk1.tampilInformasi();  
        bk1.gantiHarga(6000);  
        bk1.terjual(5);  
        bk1.hitungDiskon();  
        bk1.tampilInformasi();  
  
        Buku26 bk2 = new Buku26("Self Reward", "Maheera Ayesha", 160, 29,  
59000);  
        bk2.terjual(11);  
        bk2.hitungDiskon();  
        bk2.tampilInformasi();  
        Buku26 Rio = new Buku26("One Piece", "Eiichiro Oda", 80, 15,  
35000);  
        Rio.hitungDiskon();  
        Rio.tampilInformasi();  
    }  
}
```

## Output :

```
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 71000
Total harga penjualan Rp 0
Harga bayar setelah diskon: 0.0
=====
Terjual: 5 buku
Anda tidak mendapatkan diskon
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 8
Harga: Rp 6000
Total harga penjualan Rp 30000
Harga bayar setelah diskon: 30000.0
=====
Terjual: 11 buku
Anda mendapatkan diskon sebesar 12%
Judul: Self Reward
Pengarang: Maheera Ayesha
Jumlah halaman: 160
Sisa stok: 18
Harga: Rp 59000
Total harga penjualan Rp 649000
Harga bayar setelah diskon: 571120.0
=====
```

## 2. Buat program berdasarkan class diagram berikut ini!



Penjelasan dari atribut dan method pada class Dragon tersebut adalah sebagai berikut:

- Atribut **x** digunakan untuk menyimpan posisi koordinat x (mendatar) dari dragon, sedangkan atribut **y** untuk posisi koordinat y (vertikal)
- Atribut **width** digunakan untuk menyimpan lebar dari area permainan, sedangkan **height** untuk menyimpan panjang area
- Method **moveLeft()** digunakan untuk mengubah posisi dragon ke kiri (koordinat x akan berkurang 1), sedangkan **moveRight()** untuk bergerak ke kanan (koordinat x akan bertambah 1). Perlu diperhatikan bahwa koordinat x tidak boleh lebih kecil dari 0 atau lebih besar dari nilai width. Jika koordinat x < 0 atau x > width maka panggil method **detectCollision()**
- Method **moveUp()** digunakan untuk mengubah posisi dragon ke atas (koordinat y akan berkurang 1), sedangkan **moveDown()** untuk bergerak ke bawah (koordinat y akan bertambah nilai height. Jika koordinat y < 0 atau y > height maka panggil method **detectCollision()**
- Method **detectCollision()** akan mencetak pesan "Game Over" apabila dragon menyentuh ujung area permainan.

## Object :

```
package minggu2;

public class Dragon26 {
    int x, y, width, height;
```

```
public Dragon26(int xAwal, int yAwal, int widthArea, int heightArea) {  
    x = xAwal;  
    y = yAwal;  
    width = widthArea;  
    height = heightArea;  
}  
  
void moveLeft() {  
    x--;  
    detectCollision();  
}  
  
void moveRight() {  
    x++;  
    detectCollision();  
}  
  
void moveUp() {  
    y--;  
    detectCollision();  
}  
  
void moveDown() {  
    y++;  
    detectCollision();  
}  
  
void printPosition() {  
    System.out.println("Posisi Dragon: (" + x + ", " + y + ")");  
}  
  
void detectCollision() {  
    if (x < 0 || x > width || y < 0 || y > height) {  
        System.out.println("Dragon bergerak melewati batas!");  
        System.out.println("Game Over");  
        System.exit(0);  
    }  
}
```

```
}  
}
```

#### **Main :**

```
package minggu2;  
  
import java.util.Scanner;  
  
public class DragonMain26 {  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
        Dragon26 dragon = new Dragon26(5, 5, 10, 10);  
        dragon.printPosition();  
        boolean run = true;  
        do {  
            System.out.println("Gerakkan dragon");  
            System.out.println("W = Atas");  
            System.out.println("A = Kiri");  
            System.out.println("S = Bawah");  
            System.out.println("D = Kanan");  
            System.out.print("(W/A/S/D): ");  
            String input = scan.nextLine();  
            switch (input) {  
                case "W", "w":  
                    dragon.moveUp();  
                    dragon.printPosition();  
                    System.out.println();  
                    break;  
                case "A", "a":  
                    dragon.moveLeft();  
                    dragon.printPosition();  
                    System.out.println();  
                    break;  
                case "S", "s":  
                    dragon.moveDown();  
                    dragon.printPosition();  
                    System.out.println();  
            }  
        } while (run);  
    }  
}
```

```

        break;
    case "D", "d":
        dragon.moveRight();
        dragon.printPosition();
        System.out.println();
        break;
    default:
        System.out.println("Pilihan tidak valid");
        break;
    }
} while (run);
}
}

```

#### Output :

```

Gerakkan dragon
W = Atas
A = Kiri
S = Bawah
D = Kanan
(W/A/S/D): W
Posisi Dragon: (1, 4)

Gerakkan dragon
W = Atas
A = Kiri
S = Bawah
D = Kanan
(W/A/S/D): A
Posisi Dragon: (0, 4)

Gerakkan dragon
W = Atas
A = Kiri
S = Bawah
D = Kanan
(W/A/S/D): A
Dragon bergerak melewati batas!
Game Over

```