



## LAPORAN PRAKTIKUM JOBSHEET II OBJECT

### PRAKTIKUM

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

##### Waktu Percobaan : 50 Menit

Pada Percobaan 1 ini dilakukan pembuatan class beserta atribut dan method yang dimilikinya.

Perhatikan Class Diagram berikut ini:

Buku
judul: String pengarang: String halaman: int stok: int harga: int
tampilInformasi(): void terjual(jml: int): void restock(n: int): void gantiHarga(hrg: int): int

Berdasarkan class diagram tersebut, akan dibuat program menggunakan bahasa Java.

##### ➤ Langkah-langkah Percobaan

1. Buka text editor. Buat file baru, beri nama **Buku<NoAbsen>.java**
2. Lengkapi class **Buku** dengan atribut yang telah digambarkan di dalam class diagram tersebut

```
1 public class Buku24 {  
2     String judul, pengarang;  
3     int halaman, stok, harga;
```

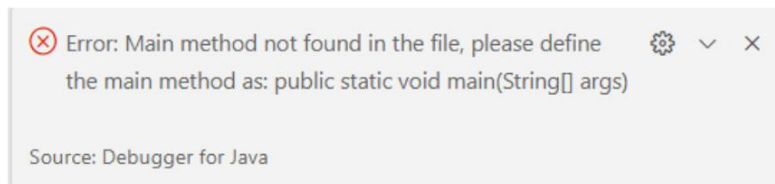
3. Lengkapi class **Buku** dengan method yang telah digambarkan di dalam class diagram tersebut.

```
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;  
}
```

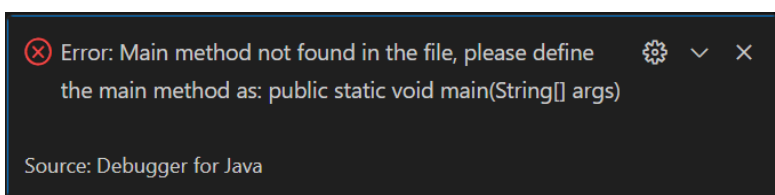
4. Compile dan run program.

➤ **Verifikasi Hasil Percobaan**

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.



Hasil running program.





### ➤ Pertanyaan

1. Sebutkan dua karakteristik class atau object!

Class atau Object memiliki dua karakteristik yaitu :

- a) Mempunyai sesuatu yang berarti bahwa objek dapat menyimpan informasi atau data yang terkait dengannya. Misalnya tentang objek "Mobil", maka atribut yang dimilikinya bisa mencakup hal-hal seperti warna mobil, tahun pembuatan, atau model mobil.
- b) Melakukan sesuatu yang berarti bahwa objek dapat melakukan tindakan atau memiliki kemampuan untuk melakukan sesuatu. Misalnya objek "Mobil" dapat memiliki metode seperti "menyalakan mesin" dan "mengerem".

2. Perhatikan class **Buku** pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Buku? Sebutkan apa saja atributnya!

Class Buku memiliki 5 atribut didalamnya yaitu judul, pengarang, halaman, stok, dan harga.

```
public class Buku24 {
    String judul, pengarang;
    int halaman, stok, harga;
}
```

3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!

Class Buku memiliki 4 method didalamnya yaitu :

- **tampilInformasi()** Menampilkan informasi lengkap tentang buku seperti judul, pengarang, jumlah halaman, sisa stok, dan harga.
- **terjual(int jml)** Mengurangkan jumlah stok buku berdasarkan jumlah penjualan (jml).
- **restock(int jml)** Menambahkan jumlah stok buku berdasarkan jumlah restok (jml).
- **gantiHarga(int hrg)** Mengubah harga buku menjadi nilai yang baru (hrg).

```
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;
}
```



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)!

```
void terjual(int jml) {
    if (stok > 0) {
        stok -= jml;
        System.out.println("Penjualan berhasil. Sisa stok: " + stok);
    } else {
        System.out.println(x:"Stok habis. Tidak dapat melakukan penjualan.");
    }
}
```

5. Mengapa method **restock()** mempunyai satu parameter berupa bilangan int?  
Metode **restock()** memiliki satu parameter berupa bilangan **int** karena parameter tersebut digunakan untuk menentukan jumlah tambahan stok yang akan ditambahkan ke buku saat proses restok dilakukan.

**Commit dan push kode program ke Github**

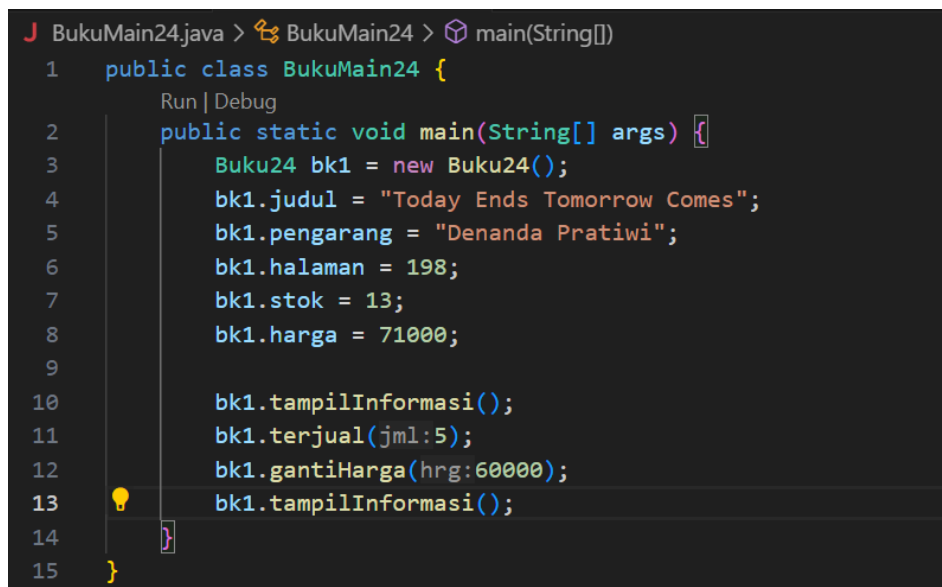
## PERCOBAAN 2: Instansiasi Object, serta Mengakses Atribut dan Method

**Waktu Percobaan: 50 Menit**

Sampai tahap ini, class **Buku** telah berhasil dibuat pada Percobaan 1. Selanjutnya, apabila class **Buku** tersebut ingin digunakan dan diakses atribut serta method-nya, maka perlu dibuat object/instance dari class **Buku** terlebih dahulu melalui proses instansiasi.

### ➤ Langkah-langkah Percobaan

1. Buat file baru, beri nama **BukuMain<NoAbsen>.java**
2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi **main()**
3. Di dalam fungsi **main()**, lakukan instansiasi, kemudian lanjutkan dengan mengakses atribut dan method dari objek yang telah terbentuk.



```

J BukuMain24.java > BukuMain24 > main(String[])
1  public class BukuMain24 {
    Run | Debug
2      public static void main(String[] args) {
3          Buku24 bk1 = new Buku24();
4          bk1.judul = "Today Ends Tomorrow Comes";
5          bk1.pengarang = "Denanda Pratiwi";
6          bk1.halaman = 198;
7          bk1.stok = 13;
8          bk1.harga = 71000;
9
10         bk1.tampilInformasi();
11         bk1.terjual(jml:5);
12         bk1.gantiHarga(hrg:60000);
13         bk1.tampilInformasi();
14     }
15 }
    
```

4. Compile dan run program.
5. **Commit dan push kode program ke Github**

### ➤ Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

```

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 60000
    
```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

howCodeDetailsInExceptionMessages' '-cp' 'C:\Users\LENOVO\AppData\
d4b008dd1bcdabb7c5df70\redhat.java\jdt_ws\Praktikum2_b8e4b498\bin'
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 71000
Penjualan berhasil. Sisa stok: 8
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 8
Harga: Rp 60000
PS D:\Algoritma Praktikum\Praktikum2>

```

### ➤ Pertanyaan

1. Pada class **BukuMain**, tunjukkan baris kode program yang digunakan untuk proses instansiasi!

```
Buku24 bk1 = new Buku24();
```

Apa nama object yang dihasilkan?

Nama objek yang dihasilkan adalah **bk1**, objek ini digunakan untuk mengakses atribut-atribut dan metode-metode yang ada dalam class Buku24.

2. Bagaimana cara mengakses atribut dan method dari suatu objek?
  - Cara mengakses atribut :  
**namaObject.namaAtribut = nilai;**
  - Cara mengakses method :  
**namaObject.namaMethod();**
3. Mengapa hasil output pemanggilan method **tampilInformasi()** pertama dan kedua berbeda?

Perbedaan hasil output pemanggilan metode **tampilInformasi()** pertama dan kedua terjadi karena adanya pemanggilan metode **terjual(5)** pada pemanggilan kedua. Pada pemanggilan pertama, informasi buku ditampilkan sesuai dengan nilai awal atribut yang telah diinisialisasi saat pembuatan objek, di mana stok masih tetap **13** dan **harga 71000**.

Setelah pemanggilan metode **terjual(5)** pada pemanggilan kedua, **nilai stok awal yaitu 13** dikurangkan dengan **nilai stok terjual yaitu 5** sehingga nilai stok berubah menjadi **8**, dan pemanggilan metode **tampilInformasi()** menampilkan informasi terkini dengan stok yang telah dikurangkan. Sehingga, perbedaan output terjadi karena terdapat aksi yang memengaruhi nilai atribut stok antara pemanggilan metode pertama dan kedua.



### PERCOBAAN 3: Membuat Konstruktor

Waktu Percobaan: 60 Menit

Pada percobaan ini, dilakukan pembuatan kode program untuk mengimplementasikan berbagai macam konstruktor berdasarkan parameternya.

#### ➤ Langkah-langkah Percobaan

1. Buka kembali class **Buku**. Tambahkan dua buah konstruktor di dalam class **Buku** tersebut, yang terdiri dari satu konstruktor default dan satu konstruktor berparameter. Konstruktor merupakan method istimewa, penempatan kode program untuk konstruktor dapat diperlakukan sama seperti method yang lain (setelah atribut).

```
public Buku24() {
}

public Buku24(String jud, String pg, int hal, int stok, int har) {
    judul = jud;
    pengarang = pg;
    halaman = hal;
    this.stok = stok;
    harga = har;
}
```

*Catatan: Apabila nama parameter sama dengan nama atribut, maka untuk merujuk pada variabel atribut ditambahkan sintaks **this** di depan nama **atribut***

2. Buka kembali class **BukuMain**. Buat sebuah object lagi bernama **bk2** dengan menggunakan konstruktor berparameter.

```
1 public class BukuMain24 {
2     public static void main(String[] args) {
3         Buku24 bk1 = new Buku24();
4         bk1.judul = "Today Ends Tomorrow Comes";
5         bk1.pengarang = "Denanda Pratiwi";
6         bk1.halaman = 198;
7         bk1.stok = 13;
8         bk1.harga = 71000;
9
10        bk1.tampilInformasi();
11        bk1.terjual(jml:5);
12        bk1.gantiHarga(hrg:60000);
13        bk1.tampilInformasi();
14
15        Buku24 bk2 = new Buku24(jud:"Self Reward", pg:"Maheera Ayesha", hal:160, stok:29, har:59000);
16        bk2.terjual(jml:11);
17        bk2.tampilInformasi();
18    }
19 }
```

4. Commit dan push kode program ke Github

### ➤ Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

```
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 60000
Judul: Self Reward
Pengarang: Maheera Ayesha
Jumlah halaman: 160
Sisa stok: 18
Harga: Rp 59000
```

```
PS D:\Algoritma Praktikum\Praktikum2> & 'C:\
howCodeDetailsInExceptionMessages' '-cp' 'C:\
d4b008dd1bcdabb7c5df70\redhat.java\jdt_ws\Pra
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 71000
Penjualan berhasil. Sisa stok: 8
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 8
Harga: Rp 60000
Penjualan berhasil. Sisa stok: 18
Judul: Self Reward
Pengarang: Maheera Ayesha
Jumlah halaman: 160
Sisa stok: 18
Harga: Rp 59000
PS D:\Algoritma Praktikum\Praktikum2>
```

### ➤ Pertanyaan

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

```
public Buku24(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);
```

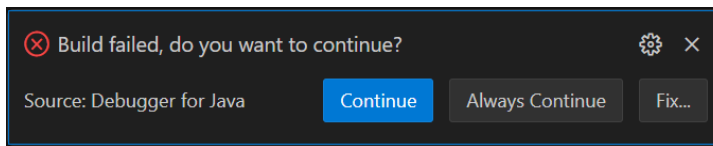
Pada class **Bukumain** baris program berikut :

```
Buku24 bk2 = new Buku24("Self Reward", "Maheera Ayesha", 160, 29, 59000);
```

Menciptakan objek baru dari class **Buku24** dengan nama **bk2**, menggunakan konstruktor berparameter untuk memberikan nilai awal pada atribut-atribut objek tersebut, seperti judul, pengarang, jumlah halaman, sisa stok, dan harga, sesuai dengan nilai yang diberikan dalam argumen konstruktor. Sehingga, objek **bk2** memiliki karakteristik yang telah ditentukan pada saat pembuatan objek.



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



Menghapus konstruktor default pada class Buku24, akan menyebabkan terjadinya kesalahan kompilasi. Hal ini terjadi karena tanpa konstruktor default, Java tidak dapat membuat objek dari class tersebut tanpa memberikan nilai awal untuk atribut-atribut.

4. Setelah melakukan instansiasi object, apakah method di dalam class **Buku** harus diakses secara berurutan? Jelaskan alasannya!

Tidak, method di dalam class Buku24 tidak harus diakses secara berurutan setelah melakukan instansiasi objek. Pemanggilan method dapat dilakukan secara terpisah dan tidak harus berurutan. Hal ini karena setiap method di dalam class Buku24 berfungsi secara independen, dan pemanggilan method tidak bergantung pada urutan atau pemanggilan method sebelumnya.

5. Buat object baru dengan nama **buku<NamaMahasiswa>** menggunakan konstruktor berparameter dari class **Buku**!

```
Buku24 bukuQusnul = new Buku24(jud:"Sagara", pg:"Tere Liye", hal:384, stok:20, har:89000);  
bukuQusnul.terjual(jml:5);  
bukuQusnul.tampilInformasi();
```

```
PS D:\Algoritma Praktikum\Praktikum2> & 'C:\Program  
howCodeDetailsInExceptionMessages' '-cp' 'C:\Users\LT  
d4b008dd1bcdabb7c5df70\redhat.java\jdt_ws\Praktikum2  
Judul: Today Ends Tomorrow Comes  
Pengarang: Denanda Pratiwi  
Jumlah halaman: 198  
Sisa stok: 13  
Harga: Rp 71000  
Penjualan berhasil. Sisa stok: 8  
Judul: Today Ends Tomorrow Comes  
Pengarang: Denanda Pratiwi  
Jumlah halaman: 198  
Sisa stok: 8  
Harga: Rp 60000  
Penjualan berhasil. Sisa stok: 18  
Judul: Self Reward  
Pengarang: Maheera Ayesha  
Jumlah halaman: 160  
Sisa stok: 18  
Harga: Rp 59000  
Penjualan berhasil. Sisa stok: 15  
Judul: Sagara  
Pengarang: Tere Liye  
Jumlah halaman: 384  
Sisa stok: 15  
Harga: Rp 89000  
PS D:\Algoritma Praktikum\Praktikum2>
```

6. Commit dan push kode program ke Github



## 2.4 Latihan Praktikum

**Waktu : 150 Menit**

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

Class diagram **Buku** setelah penambahan ketiga method tersebut adalah sebagai berikut.

Buku
judul: String pengarang: String halaman: int stok: int harga: int
tampilInformasi(): void terjual(jml: int): void restock(n: int): void gantiHarga(hrg: int): int <b>hitungHargaTotal(): int</b> <b>hitungDiskon(): int</b> <b>hitungHargaBayar(): int</b>



## Code Program Latihan 1

```
J BukuMain24.java > BukuMain24 > main(String[])
1  public class BukuMain24 {
    Run | Debug
2  public static void main(String[] args) {
3      Buku24 bk1 = new Buku24();
4      bk1.judul = "Today Ends Tomorrow Comes";
5      bk1.pengarang = "Denanda Pratiwi";
6      bk1.halaman = 198;
7      bk1.stok = 13;
8      bk1.harga = 71000;
9      bk1.tampilInformasi();
10
11     bk1.gantiHarga(hrg:60000);
12     System.out.println(x:"\nInformasi Buku Setelah Mengganti Harga:");
13     bk1.tampilInformasi();
14
15     System.out.println(x:"\nInformasi Buku Sebelum Terjual:");
16     bk1.tampilInformasi();
17
18     int jumlahBukuTerjual = 5;
19     int hargaTotal = bk1.hitungHargaTotal(jumlahBukuTerjual);
20     int diskon = bk1.hitungDiskon(hargaTotal);
21     int hargaBayar = bk1.hitungHargaBayar(jumlahBukuTerjual);
22
23     System.out.println(x:"\nSetelah Terjual 5 Buku:");
24     System.out.println("Harga Total: " + hargaTotal);
25     System.out.println("Diskon: " + diskon);
26     System.out.println("Harga Bayar: " + hargaBayar);
27
28     Buku24 bk2 = new Buku24(jud:"Self Reward", pg:"Maheera Ayesha", hal:160, stok:29, har:59000);
29     bk2.terjual(jml:11);
30
31     System.out.println(x:"\nInformasi Buku bk2 (Setelah Terjual 11 Buku:)");
32     int hargaTotalBk2 = bk2.hitungHargaTotal(jml:11);
33     int diskonBk2 = bk2.hitungDiskon(hargaTotalBk2);
34     int hargaBayarBk2 = bk2.hitungHargaBayar(jml:11);
35
36     System.out.println("Harga Total: " + hargaTotalBk2);
37     System.out.println("Diskon: " + diskonBk2);
38     System.out.println("Harga Bayar: " + hargaBayarBk2);
39     bk2.tampilInformasi();
40
41     Buku24 bukuQusnul = new Buku24(jud:"Sagara", pg:"Tere Liye", hal:384, stok:20, har:89000);
42     bukuQusnul.terjual(jml:5);
43
44     System.out.println(x:"\nInformasi Buku bukuQusnul (Setelah Terjual 5 Buku:)");
45     int hargaTotalQusnul = bukuQusnul.hitungHargaTotal(jml:5);
46     int diskonQusnul = bukuQusnul.hitungDiskon(hargaTotalQusnul);
47     int hargaBayarQusnul = bukuQusnul.hitungHargaBayar(jml:5);
48
49     System.out.println("Harga Total: " + hargaTotalQusnul);
50     System.out.println("Diskon: " + diskonQusnul);
51     System.out.println("Harga Bayar: " + hargaBayarQusnul);
52     bukuQusnul.tampilInformasi();
53 }
54 }
```



## Hasil Running Latihan 1

```
PS D:\Algoritma Praktikum\Praktikum2> & 'C:\Program Files\Java\
howCodeDetailsInExceptionMessages' '-cp' 'C:\Users\LENOVO\AppData
d4b008dd1bcdabb7c5df70\redhat.java\jdt_ws\Praktikum2_b8e4b498\b
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 71000

Informasi Buku Setelah Mengganti Harga:
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 60000

Informasi Buku Sebelum Terjual:
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 60000

Setelah Terjual 5 Buku:
Harga Total: 300000
Diskon: 36000
Harga Bayar: 264000
Penjualan berhasil. Sisa stok: 18
```

```
Informasi Buku bk2 (Setelah Terjual 11 Buku):
Harga Total: 649000
Diskon: 77880
Harga Bayar: 571120
Judul: Self Reward
Pengarang: Maheera Ayesha
Jumlah halaman: 160
Sisa stok: 18
Harga: Rp 59000
Penjualan berhasil. Sisa stok: 15

Informasi Buku bukuQusnul (Setelah Terjual 5 Buku):
Harga Total: 445000
Diskon: 53400
Harga Bayar: 391600
Judul: Sagara
Pengarang: Tere Liye
Jumlah halaman: 384
Sisa stok: 15
Harga: Rp 89000
PS D:\Algoritma Praktikum\Praktikum2> █
```



2. Buat program berdasarkan class diagram berikut ini!

Dragon
x: int y: int width: int height: int
moveLeft(): void moveRight(): void moveUp(): void moveDown(): void printPosition(): void detectCollision(x: int, y: int): void

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  $x > \text{width}$  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 1). Perlu diperhatikan bahwa koordinat y tidak boleh lebih kecil dari 0 atau lebih besar dari nilai height. Jika koordinat  $y < 0$  atau  $y > \text{height}$  maka panggil method **detectCollision()**
- Method **detectCollision()** akan mencetak pesan "Game Over" apabila dragon menyentuh ujung area permainan.



## Code Program Latihan 2

```
J Dragon24.java > ⚙ Dragon24 > 🏠 main(String[])
1  public class Dragon24 {
2      private int x, y, width, height;
3      public Dragon24(int initialX, int initialY, int gameWidth, int gameHeight) {
4          x = initialX;
5          y = initialY;
6          width = gameWidth;
7          height = gameHeight;
8      }
9
10     public void moveLeft() {
11         x--;
12         checkCollision();
13     }
14
15     public void moveRight() {
16         x++;
17         checkCollision();
18     }
19
20     public void moveUp() {
21         y--;
22         checkCollision();
23     }
24
25     public void moveDown() {
26         y++;
27         checkCollision();
28     }
29
30     public void printPosition() {
31         System.out.println("Dragon Position - X: " + x + ", Y: " + y);
32     }
33
34     private void checkCollision() {
35         if (x < 0 || x >= width || y < 0 || y >= height) {
36             detectCollision();
37         }
38     }
39
40     private void detectCollision() {
41         System.out.println(x: "Game Over");
42         System.exit(status:0);
43     }
44
45     Run | Debug
46     public static void main(String[] args) {
47         Dragon24 dragon = new Dragon24(initialX:5, initialY:5, gameWidth:10, gameHeight:10);
48         dragon.moveRight();
49         dragon.printPosition();
50
51         dragon.moveDown();
52         dragon.printPosition();
53
54         dragon.moveLeft();
55         dragon.printPosition();
56
57         dragon.moveUp();
58         dragon.printPosition();
59     }
```



## Hasil Running Latihan 2

```
PS D:\Algoritma Praktikum\Praktikum2> & 'C:\Program Files\Java\
howCodeDetailsInExceptionMessages' '-cp' 'C:\Users\LENOVO\AppData
d4b008dd1bcdabb7c5df70\redhat.java\jdt_ws\Praktikum2_b8e4b498\bi
Dragon Position - X: 6, Y: 5
Dragon Position - X: 6, Y: 6
Dragon Position - X: 5, Y: 6
Dragon Position - X: 5, Y: 5
PS D:\Algoritma Praktikum\Praktikum2>
```

Link Repository Github :

<https://github.com/qusnuldiah/Algoritma-StrukturData/tree/main/Praktikum2>