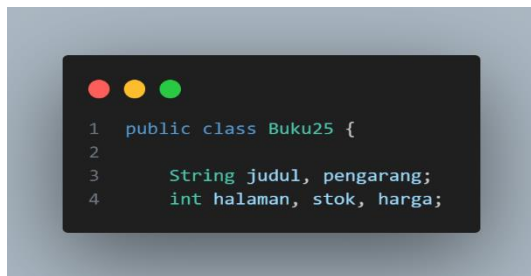


Percobaan 01



```
1 public class Buku25 {
2
3     String judul, pengarang;
4     int halaman, stok, harga;
```

Kelas Buku25 mendefinisikan struktur data untuk menyimpan informasi tentang sebuah buku, seperti judul, pengarang, jumlah halaman, stok, dan harga. Dan deklarasi String (judul, pengarang) int (halaman, stok, harga)



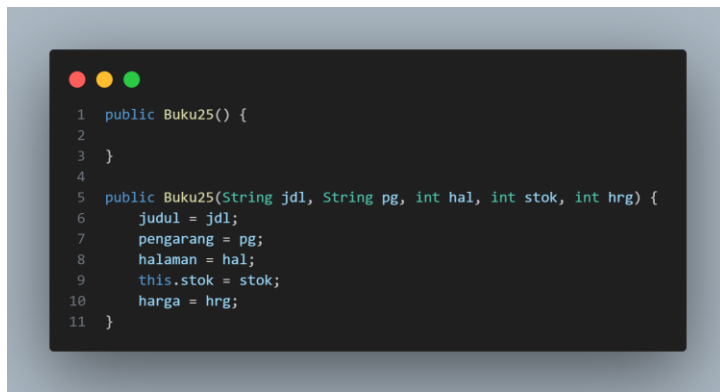
```
1 void tampilInformasi() {
2     System.out.println("Judul: " + judul);
3     System.out.println("Pengarang: " + pengarang);
4     System.out.println("Jumlah halaman: " + halaman);
5     System.out.println("Sisa stok: " + stok);
6     System.out.println("Harga: Rp " + harga);
```

Metode tampilInformasi membantu dalam menampilkan informasi buku seperti label judul, pengarang, halaman, stok, harga.



```
1 }
2
3 void terjual (int jml) {
4     stok = stok - jml;
5 }
6
7 void restock(int jml) {
8     stok = stok + jml;
9 }
10
11 void gantiHarga(int hrg) {
12     harga = hrg;
13 }
14 }
15
```

mengimplementasikan tiga fungsi yang berkaitan dengan stok dan harga



Konstruktor Buku25(String jdl, String pg, int hal, int stok, int hrg) konstruktor yang menerima beberapa parameter. Ketika objek Buku25 dibuat dengan menyediakan nilai untuk semua parameter ini, konstruktor ini akan dipanggil untuk menginisialisasi atribut-atribut buku dengan nilai-nilai yang diberikan.

2.1.3 Pertanyaan

1. Sebutkan dua karakteristik class atau object!

- Encapsulation: Class atau objek dapat menggabungkan data (atribut) dan metode (fungsi) ke dalam satu kesatuan, sehingga menyembunyikan detail implementasi dari luar dan mendorong pemisahan tanggung jawab.

-Inheritance: Class dapat mewarisi sifat dan perilaku dari class lain, memungkinkan untuk membangun hierarki class dan menghindari duplikasi kode.

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

- judul (title), pengarang (author), halaman (number of pages), stok (stock)

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

-tampilinformasi(): Menampilkan informasi tentang buku.

-terjual(int jml): Mengurangi stok buku saat terjadi penjualan.

-restock(int jml): Menambah stok buku saat proses restocking.

-gantiHarga(int hrg): Mengubah harga buku.

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)!



5. Menurut Anda, mengapa method restock() mempunyai satu parameter berupa bilangan int?

Parameter jml (jumlah) dalam metode restock mewakili jumlah buku yang akan ditambahkan ke stok. Dengan adanya parameter ini, metode tersebut menjadi lebih fleksibel, memungkinkan pemanggil untuk menentukan berapa banyak buku yang harus di-restock setiap kali metode dipanggil. Parameterisasi ini memungkinkan penyesuaian dinamis dari jumlah restocking berdasarkan kebutuhan aplikasi.

6. Commit dan push kode program ke Github

Percobaan 02

```
1 public static void main(String[] args) {  
2     Buku25 bk1 = new Buku25();  
3     bk1.judul = "Today Ends Tomorrow Comes";  
4     bk1.pengarang = "Denanda Pratiwi";  
5     bk1.halaman = 198;  
6     bk1.stok = 13;  
7     bk1.harga = 71000;  
}
```

membuat objek bk1 dari kelas Buku25 dan menginisialisasi atribut-atributnya dengan nilai-nilai tertentu.

```
1 int jml, hrg;
```

Deklarasi variabel jml dan hrg yang akan digunakan nanti.

```
1 bk1.tampilInformasi();  
2 bk1.terjual(jml=5);  
3 bk1.gantiHarga(hrg=60000);  
4 bk1.tampilInformasi();  
5
```

Memanggil metode tampilInformasi() dari objek bk1 untuk menampilkan informasi buku.

```
1 Buku25 bk2 = new Buku25(jdl="Self Reward", pg="Maheera Ayesha", hal=160, stok=29, hrg=59000);  
2 bk2.terjual(jml=11);  
3 bk2.tampilInformasi();
```

Membuat objek bk2 dari kelas Buku25 dengan menggunakan konstruktor yang menerima beberapa parameter. Nilai-nilai ini akan digunakan untuk menginisialisasi atribut-atribut objek bk2

2.2.3 Pertanyaan

1. Pada class BukuMain, tunjukkan baris kode program yang digunakan untuk proses instansiasi! Apa nama object yang dihasilkan? `Buku13 bk1 = new Buku13();` Dan Objek yang dihasilkan adalah `bk1`

2. Bagaimana cara mengakses atribut dan method dari suatu objek? menggunakan operator titik (.). Contohnya, pada baris kode program `bk1.tampilInformasi();` Kode di atas mengakses method `tampilInformasi()` dari objek `bk1`.

3. Mengapa hasil output pemanggilan method `tampilInformasi()` pertama dan kedua berbeda? Berbeda, karena pada pemanggilan pertama, informasi buku ditampilkan sebelum ada penjualan (terjual(5)) dan perubahan harga (`gantiHarga(60000)`), sedangkan pada pemanggilan kedua, informasi buku ditampilkan setelah ada penjualan dan perubahan harga. Jadi, hasil output mencerminkan kondisi terbaru dari objek `bk1`.

```
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 71000
5 buku terjual. Sisa stok: 8
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 8
Harga: Rp 60000
```

Percobaan 3

```
1 Buku25 bukuDenanda = new Buku25 ("Today Ends Tomorrow Come", "Denanda Pratiwi", 198, 13, 71000);
2   bukuDenanda.tampilInformasi();
3
4 Buku25 bukuMaheera = new Buku25 ("Self Reward", "Maheera Ayesha", 160, 29, 59000);
5   bukuMaheera.tampilInformasi();
```

Baris ini membuat objek `bukuDenanda` dari kelas `Buku25` dengan menggunakan konstruktor berparameter. Objek ini mewakili buku dengan judul "Today Ends Tomorrow Come", pengarang "Denanda Pratiwi", jumlah halaman 198, stok 13, dan harga 71000. Dan membuat objek `bukuMaheera` dari kelas `Buku25` dengan menggunakan konstruktor berparameter. Objek ini mewakili buku dengan judul "Self Reward", pengarang "Maheera Ayesha", jumlah halaman 160, stok 29, dan harga 59000. Lalu Memanggil metode `tampilInformasi()` dari objek buku keduanya untuk menampilkan informasi tentang buku Maheera dan denada

2.3.3 Pertanyaan

1. Pada class Buku di Percobaan 3, tunjukkan baris kode program yang digunakan untuk

```
public Buku25(String jdl, String pg, int hal, int stok, int hrg) {
    judul = jdl;
    pengarang = pg;
    halaman = hal;
    this.stok = stok;
    harga = hrg;
}
```

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 baris ini, objek bk2 dibuat menggunakan konstruktor berparameter dari class Buku13. Ini menunjukkan cara menggunakan konstruktor berparameter untuk membuat dan menginisialisasi objek Buku13 sekaligus.

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

Hasilnya akan menjadi kesalahan kompilasi karena tidak ada konstruktor default yang dapat digunakan untuk membuat objek Buku13. Program perlu menyediakan nilai untuk setiap parameter konstruktor

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

Tidak, karena Masing-masing method berfungsi secara independen dan dapat dipanggil secara terpisah tanpa perlu mengikuti urutan tertentu.

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

```
public class BukuMahasiswa {
    Run | Debug
    public static void main(String[] args) {
        Buku25
        Buku25 bukuDenanda = new Buku25 (jdl:"Today Ends Tomorrow Come", pg:"Denanda Pratiwi", h
        bukuDenanda.tampilInformasi();

        Buku25 bukuMaheera = new Buku25 (jdl:"Self Reward", pg:"Maheera Ayesha", hal:160, stok:2
        bukuMaheera.tampilInformasi();
    }
}
```

6. Commit dan push kode program ke Github

Latihan 1

```
1 int hitungHargaTotal(int jmlBukuTerjual) {
2     return harga * jmlBukuTerjual;
}
```

Method ini menghitung harga total dengan mengalikan harga per buku dengan jumlah buku yang terjual (jmlBukuTerjual).

```
1 double hitungDiskon(int hargaTotal) {  
2     double diskon = 0;  
3     if (hargaTotal > 150000) {  
4         diskon = 0.12;  
5     } else if (hargaTotal >= 75000 && hargaTotal <= 150000) {  
6         diskon = 0.05;  
7     }  
8     return diskon;  
}
```

Method ini menghitung jumlah diskon berdasarkan harga total yang diberikan sebagai argumen. Jika harga total lebih dari 150000, diskon adalah 12%. Jika harga total antara 75000 sampai 150000, diskon adalah 5%. Jika harga total kurang dari 75000, tidak ada diskon.

```
1 double hitungHargaBayar(int hargaTotal) {  
2     double diskon = hitungDiskon(hargaTotal);  
3     return hargaTotal - (hargaTotal * diskon);  
}
```

Method ini menghitung harga bayar setelah diskon. Menggunakan method `hitungDiskon()` untuk menghitung diskon yang diberikan, kemudian mengurangi diskon dari harga total untuk mendapatkan harga bayar akhir.

Latihan 2

```
1 public class Dragon25 {
2     private int x;
3     private int y;
4     private int width;
5     private int height;
6
7     public Dragon25(int x, int y, int width, int height) {
8         this.x = x;
9         this.y = y;
10        this.width = width;
11        this.height = height;
12    }
13
14    public void moveLeft() {
15        if (x - 1 >= 0) {
16            x--;
17        } else {
18            detectCollision(x - 1, y);
19        }
20    }
21
22    public void moveRight() {
23        if (x + 1 < width) {
24            x++;
25        } else {
26            detectCollision(x + 1, y);
27        }
28    }
29
30    public void moveUp() {
31        if (y - 1 >= 0) {
32            y--;
33        } else {
34            detectCollision(x, y - 1);
35        }
36    }
37
38    public void moveDown() {
39        if (y + 1 < height) {
40            y++;
41        } else {
42            detectCollision(x, y + 1);
43        }
44    }
45
46    public void detectCollision(int x, int y) {
47        System.out.println("Game Over");
48    }
49
50    public void printPosition() {
51        System.out.println("Dragon position: (" + x + ", " + y + ")");
52    }
53 }
54 }
```

```
1 public class DragonMain25 {
2     public static void main(String[] args) {
3
4         Dragon25 dragon = new Dragon25(5, 5, 10, 10);
5
6
7         dragon.printPosition();
8
9         dragon.moveLeft();
10        dragon.printPosition();
11
12        dragon.moveUp();
13        dragon.printPosition();
14
15        dragon.moveRight();
16        dragon.printPosition();
17
18        dragon.moveDown();
19        dragon.printPosition();
20
21        dragon.moveLeft();
22        dragon.printPosition();
23    }
24 }
25 }
```