

JOBSHEET 2

Object



Name

Sherly Lutfi Azkiah Sulistyawati

NIM

2341720241

Class

1I

Major

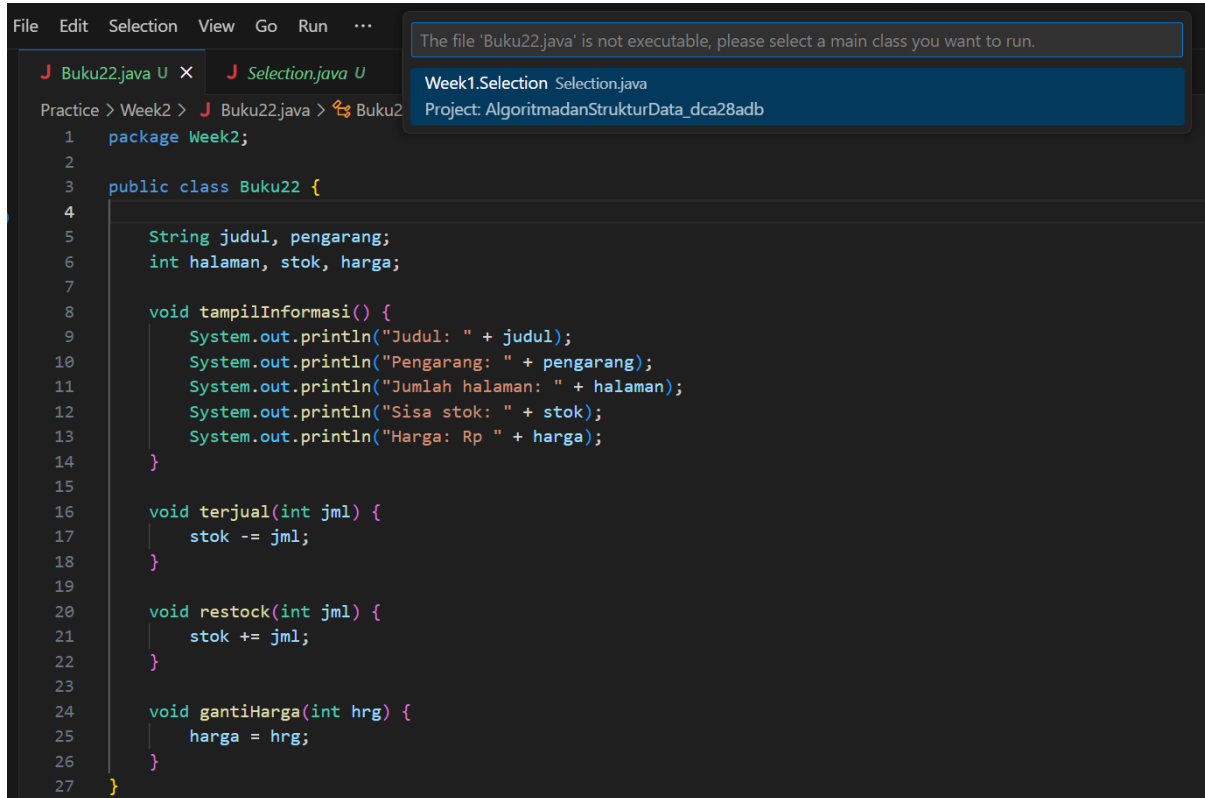
Information Technology

Study Program

D4 Informatics Engineering

Percobaan 1: Deklarasi Class, Atribut dan Method

Verifikasi Hasil Percobaan



The screenshot shows an IDE with a Java file named 'Buku22.java'. The code defines a class 'Buku22' with attributes 'judul', 'pengarang', 'halaman', 'stok', and 'harga'. It also includes four methods: 'tampilInformasi()', 'terjual()', 'restock()', and 'gantiHarga()'. A message box at the top right states: 'The file 'Buku22.java' is not executable, please select a main class you want to run.' Below this message, a dropdown menu shows 'Week1.Selection Selection.java' and 'Project: AlgoritmadanStrukturData_dca28adb'.

```
1 package Week2;
2
3 public class Buku22 {
4
5     String judul, pengarang;
6     int halaman, stok, harga;
7
8     void tampilInformasi() {
9         System.out.println("Judul: " + judul);
10        System.out.println("Pengarang: " + pengarang);
11        System.out.println("Jumlah halaman: " + halaman);
12        System.out.println("Sisa stok: " + stok);
13        System.out.println("Harga: Rp " + harga);
14    }
15
16    void terjual(int jml) {
17        stok -= jml;
18    }
19
20    void restock(int jml) {
21        stok += jml;
22    }
23
24    void gantiHarga(int hrg) {
25        harga = hrg;
26    }
27 }
```

Pertanyaan

1. Sebutkan dua karakteristik class atau object!

Object memiliki 2 karakteristik, yaitu **mempunyai sesuatu** (istilahnya data, properti, variable, state, dan atribut) dan **melakukan sesuatu** (istilahnya tingkah laku, behaviour, fungsi, dan method). Class masih berupa rancangan yang bersifat umum. Sedangkan object merupakan bentuk nyata yang terbentuk dari suatu class dan memiliki sifat spesifik.

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

Buku memiliki 5 atribut. Atributnya adalah judul, pengarang, halaman, stok, dan harga.

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

Ada 4 method, yaitu tampilInformasi, terjual, restock, dan gantiHarga.

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;  
    }  
}
```

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

Method `restock()` memiliki 1 parameter karena kita membutuhkan return stok akhir buku setelah melewati method `restock`.

6. Commit dan push kode program ke Github

Percobaan 2: Instansiasi Object, serta Mengakses Atribut dan Method

Verifikasi Hasil Percobaan

```
public class BukuMain22 {  
  
    Run | Debug  
    public static void main(String[] args) {  
  
        Buku22 bk1 = new Buku22();  
        bk1.judul = "Todays Ends Tomorrow Comes";  
        bk1.pengarang = "Denanda Pratiwi";  
        bk1.halaman = 198;  
        bk1.stok = 13;  
        bk1.harga = 71000;  
  
        bk1.tampilInformasi();  
        bk1.terjual(jml:5);  
        bk1.gantiHarga(hrg:60000);  
        bk1.tampilInformasi();  
    }  
}
```

```
Judul: Todays Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 71000
Judul: Todays Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 8
Harga: Rp 60000
PS D:\College\Semester 2\AlgoritmadanStrukturData>
```

Pertanyaan

1. Pada class BukuMain, tunjukkan baris kode program yang digunakan untuk proses instansiasi! Apa nama object yang dihasilkan?

```
Buku22 bk1 = new Buku22();
```

Object yang dihasilkan adalah bk1.

2. Bagaimana cara mengakses atribut dan method dari suatu objek?

Cara mengakses atribut:

```
bk1.judul = "Todays Ends Tomorrow Comes";
bk1.pengarang = "Denanda Pratiwi";
bk1.halaman = 198;
bk1.stok = 13;
bk1.harga = 71000;
```

Cara mengakses method:

```
bk1.tampilInformasi();
bk1.terjual(jml:5);
bk1.gantiHarga(hrg:60000);
bk1.tampilInformasi();
```

3. Mengapa hasil output pemanggilan method tampilInformasi() pertama dan kedua berbeda?

Karena terdapat perubahan nilai pada method terjual() dan method gantiHarga().

Percobaan 3: Membuat Konstruktor

Verifikasi Hasil Percobaan

```
public Buku22() {  
  
}  
  
public Buku22(String jud, String pg, int hal, int stok, int har) {  
    judul = jud;  
    pengarang = pg;  
    halaman = hal;  
    this.stok = stok;  
    harga = har;  
}
```

```
Buku22 bk2 = new Buku22(jud:"Self Reward", pg:"Maheera Ayesha", hal:160, stok:29, har:59000);  
bk2.terjual(jml:11);  
bk2.tampilInformasi();
```

```
Judul: Todays Ends Tomorrow Comes  
Pengarang: Denanda Pratiwi  
Jumlah halaman: 198  
Sisa stok: 13  
Harga: Rp 71000  
Judul: Todays 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
```

Pertanyaan

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

```
public Buku22(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);
```

Membuat object baru bernama bk2 dengan menggunakan konstruktor berparameter.

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

Hasilnya akan error, karena pada object bk1 menggunakan konstruktor default. Jika dihapus konstruktor pada object bk1 tidak terdefinisi.

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

Method pada class Buku tidak harus diakses secara berurutan karena class tersebut hanya template . Yang diakses secara berurutan adalah class BukuMain.

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

```
Buku22 bukuSherly = new Buku22(jud:"One of Us Is Lying", pg:"Karen M. McManus", hal:368, stok:15,
bukuSherly.terjual(jml:5);
bukuSherly.tampilInformasi());
```

6. Commit dan push kode program ke Github

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

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 hitungHargaTotal(): int hitungDiskon(): int hitungHargaBayar(): int

Answer

```
int hitungHargaTotal(int jml) {  
    return harga * jml;  
}  
  
int hitungDiskon(int jml) {  
    int diskon = 0;  
    int hargaTotal = hitungHargaTotal(jml);  
    if (hargaTotal > 150000)  
        diskon = (int)(0.12*hargaTotal);  
    else if (hargaTotal < 150000 && hargaTotal > 75000)  
        diskon = (int)(0.05*hargaTotal);  
    else  
        diskon = 0;  
    return diskon;  
}  
  
int hitungHargaBayar(int jml) {  
    return hitungHargaTotal(jml) - hitungDiskon(jml);  
}
```

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.

Answer

```
Practice > Week2 > J Dragon.java > ⚙ Dragon > 📄 moveup()
1  package Week2;
2
3  public class Dragon {
4      int width, height, x, y;
5      public Dragon(int w, int h){
6          width = w;
7          height = h;
8          x = w/2;
9          y = h/2;
10     }
11     void printPosition(){
12         System.out.printf(format:"(%d, %d)",x,y);
13         System.out.println();
14     }
15     void moveleft(){
16         if(x>0)
17             x--;
18         printPosition();
19         detectColission();
20     }
21     void moveright(){
22         if(x<width)
23             x++;
24         printPosition();
25         detectColission();
26     }
```






```

27     void moveup(){
28         if(y>0)
29             y--;
30         printPosition();
31         detectColission();
32     }
33     void movedown(){
34         if(y<height)
35             y++;
36         printPosition();
37         detectColission();
38     }
39     void detectColission(){
40         if(x<=0 || y<=0 || x>=width || y>=height)
41             System.out.println(x:"Game Over!");
42     }
43     Run | Debug
44     public static void main(String[] args) {
45         Dragon p = new Dragon(w:50, h:100);
46         for(int i=0;i<=50;i++){
47             p.movedown();
48         }
49     }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

 Run: Dragon + v   ...

```

(25, 71)
(25, 72)
(25, 73)
(25, 74)
(25, 75)
(25, 76)
(25, 77)
(25, 78)
(25, 79)
(25, 80)
(25, 81)
(25, 82)
(25, 83)
(25, 84)
(25, 85)
(25, 86)
(25, 87)
(25, 88)
(25, 89)
(25, 90)
(25, 91)
(25, 92)
(25, 93)
(25, 94)
(25, 95)
(25, 96)
(25, 97)
(25, 98)
(25, 99)
(25, 100)
Game Over!
(25, 100)
Game Over!

```