

LAPORAN PRAKTIKUM

ALGORITMA DAN STRUKTUR DATA

JOBSHEET II – OBJECT

NAMA : KAYLA RACHMAUDINA SATITI PUTRI

NIM : 2341760103

PRAKTIKUM 1 - Deklarasi Class, Atribut dan Method



```
1  public class buku15 {
2      String judul, pengarang;
3      int halaman, stok, harga;
4
5      void tampilInformasi() {
6          System.out.println("\nJudul          : " + judul);
7          System.out.println("Pengarang       : " + pengarang);
8          System.out.println("Jumlah halaman : " + halaman);
9          System.out.println("Sisa stok      : " + stok);
10         System.out.println("Harga          : " + harga);
11     }
12
13     void terjual(int jml) {
14         if (stok > 0) {
15             stok -= jml;
16         }
17     }
18
19     void restock(int jml) {
20         stok += jml;
21     }
22
23     void gantiHarga(int hrg) {
24         harga = hrg;
25     }
26 }
```

Verivikasi hasil percobaan :

The file 'Buku15.java' is not executable, please select a main class you want to run.

Di sini terlihat bahwa program di atas tidak bisa dijalankan dikarenakan tidak adanya fungsi main.

Pertanyaan

1. Sebutkan dua karakteristik class atau object!
 - Melakukan sesuatu dan mempunyai sesuatu
2. Perhatikan class Buku pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Buku? Sebutkan apa saja atributnya!
 - Ada 5 atribut yaitu judul, pengarang, halaman, stok, harga.
3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!
 - Ada 4 method yaitu tampilInformasi(), terjual(int jml), restock(int jml), gantiHarga(int 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)!

```
25 void terjual(int jml) {  
26     if (stok > 0) {  
27         stok -= jml;  
28     }  
29 }
```

5. Menurut Anda, mengapa method restock() mempunyai satu parameter berupa bilangan int? Digunakan untuk menentukan jumlah buku yang akan ditambahkan ke stok.
6. Commit dan push kode program ke Github

Kaylarsp percobaan 1		ec20315 · 2 minutes ago	History
Name	Last commit message	Last commit ...	
..			
Buku15.java	percobaan 1	2 minutes ago	

PRAKTIKUM 2 - Instansiasi Object, serta Mengakses Atribut dan Method

```
1 public class bukuMain15 {
2     public static void main(String[] args) {
3         buku15 bk1 = new buku15();
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(5);
12        bk1.gantiHarga(60000);
13        bk1.tampilInformasi();
14    }
15 }
```

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

Judul      : Today Ends Tomorrow Comes
Pengarang   : Denanda Pratiwi
Jumlah halaman : 198
Sisa stok   : 8
Harga       : 60000
```

Pertanyaan

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

```
1 buku15 bk1 = new buku15();
```

Nama objek yang dihasilkan adalah bk1

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

```
bk1.tampilInformasi();  
bk1.terjual(5);  
bk1.gantiHarga(60000);  
bk1.tampilInformasi();
```

Dengan namaObjek.namaMethod()

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

Karena pada pemanggilan pertama, stok buku terjual sebanyak 5 buah. Jadi, otomatis pada pemanggilan kedua stok buku akan berkurang sebanyak lima buah sehingga menjadi 8 buah.

PRAKTIKUM 3 – Membuat Konstruktor

```
1 public buku15() {  
2  
3 }  
4  
5 public buku15(String jud, String pg, int hal, int stok, int har) {  
6     judul = jud;  
7     pengarang = pg;  
8     halaman = hal;  
9     this.stok = stok; // menggunakan this karena parameter sm atribut sm  
10    harga = har;  
11 }
```

```
1 // menambahkan objek bk2 pk konstruktor berparameter  
2 buku15 bk2 = new buku15("Self Reward", "Maheera Ayesha", 160, 29, 59000);  
3 bk2.terjual(11);  
4 bk2.tampilInformasi();
```

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

```
Judul      : Today Ends Tomorrow Comes  
Pengarang  : Denanda Pratiwi  
Jumlah halaman : 198  
Sisa stok  : 8  
Harga      : 60000
```

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

Pertanyaan

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

```
5 public buku15(String jud, String pg, int hal, int stok, int har) {  
6     judul = jud;  
7     pengarang = pg;  
8     halaman = hal;  
9     this.stok = stok; // menggunakan this karena parameter sm atribut sm  
10    harga = har;  
11 }
```

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

```
1 // menambahkan objek bk2 pk konstruktor berparameter  
2 buku15 bk2 = new buku15("Self Reward", "Maheera Ayesha", 160, 29, 59000);
```

Baris kode di atas menunjukkan proses instansiasi objek bk2 dari class buku15. Parameter seperti ("Self Reward", "Maheera Ayesha", 160, 29, 59000) akan dikirimkan ke konstruktor buku15 untuk menginisialisasi atribut dari objek yang baru saja dibuat.

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

```
4 buku15 bk1 = new buku15();  
✖ BukuMain15.java 1 of 1 problem  
The constructor buku15() is undefined Java(134217858)
```

Terjadi error dikarenakan program tidak dapat menemukan cara untuk membuat objek dari class tersebut. Hal ini terjadi karena program membutuhkan konstruktor untuk menginisialisasi atribut-atribut dari objek yang baru saja dibuat.

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

Tidak harus secara berurutan, pemanggilan method dapat dipanggil secara bebas dan terpisah dikarenakan method memang tidak terkait satu sama lain dalam pemanggilannya. Pemanggilan method disesuaikan dengan kebutuhan user.

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

```
1 // membuat objek baru dg nama saya
2 buku15 bukuKayla = new buku15("Tentang Kamu", "Tere Liye", 503, 30, 85000);
3 bukuKayla.tampilInformasi();
4 bukuKayla.terjual(23);
5 bukuKayla.tampilInformasi();
6 bukuKayla.restock(100);
7 bukuKayla.tampilInformasi();
8 }
```

```
Judul      : Tentang Kamu
Pengarang  : Tere Liye
Jumlah halaman : 503
Sisa stok  : 30
Harga      : 85000
```

```
Judul      : Tentang Kamu
Pengarang  : Tere Liye
Jumlah halaman : 503
Sisa stok  : 7
Harga      : 85000
```

```
Judul      : Tentang Kamu
Pengarang  : Tere Liye
Jumlah halaman : 503
Sisa stok  : 107
Harga      : 85000
```

```
d:\DASPRO-JAVA>
```

Latihan Praktikum

1. Pada class Buku yang telah dibuat, tambahkan tiga method yaitu hitungHargaTotal(), hitungDiskon(), dan hitungHargaBayar()

```
1      void hitungHargaTotal(int jml) {
2          hargaTotal = harga * jml;
3      }
4
5      void hitungDiskon() {
6          if (hargaTotal > 150000) {
7              diskon = hargaTotal * 0.12;
8          } else if (hargaTotal >= 75000 && hargaTotal <= 150000) {
9              diskon = hargaTotal * 0.05;
10         }
11     }
12
13     double hargaBayar;
14
15     void hitungHargaBayar() {
16         hitungDiskon();
17         hargaBayar = hargaTotal - diskon;
18     }
```

```
1      // membuat objek baru dg nama saya
2      buku15 bukuKayla = new buku15("Tentang Kamu", "Tere Liye", 503, 30, 85000);
3      bukuKayla.terjual(5);
4      bukuKayla.hitungHargaTotal(5);
5      bukuKayla.hitungDiskon();
6      bukuKayla.hitungHargaBayar();
7      bukuKayla.tampilInformasi();
8
```

```
Judul      : Tentang Kamu
Pengarang   : Tere Liye
Jumlah halaman : 503
Sisa stok   : 25
Harga total : Rp.425000.0
Harga diskon : Rp.51000.0
Harga bayar : Rp.374000.0
```

2. Membuat program dragon

```
1  public class dragon15 {
2      int x, y, width, height;
3
4      // membuat konstruktor
5      public dragon15(int w, int h) {
6          width = w;
7          height = h;
8      }
9
10     void moveLeft() {
11         if (x > 0 && x <= width) {
12             x--;
13         } else {
14             detectCollision();
15         }
16     }
17
18     void moveRight() {
19         if (x >= 0 && x < width) {
20             x++;
21         } else {
22             detectCollision();
23         }
24     }
25
26     void moveUp() {
27         if (y > 0 && y <= height) {
28             y--;
29         } else {
30             detectCollision();
31         }
32     }
33
34     void moveDown() {
35         if (y >= 0 && y < height) {
36             y++;
37         } else {
38             detectCollision();
39         }
40     }
41
42     void detectCollision() {
43         if (x < 0 || x > width || y < 0 || y > height) {
44             System.out.println("Game Over");
45             System.exit(0);
46         }
47     }
48
49     void printPosition() {
50         System.out.printf("x : %d\n y : %d", x, y);
51     }
52 }
```



```

1  import java.util.Scanner;
2
3  public class dragonMain15 {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          dragon15 dragon = new dragon15(3, 5);
7
8          while (true) {
9              System.out.println();
10             System.out.println("\nMove dragon");
11             System.out.println("1. Move right\n2. Move left\n3. Move up\n4. Move down\n0. Exit");
12             System.out.print("\nEnter number : ");
13             int number = sc.nextInt();
14
15             System.out.println();
16
17             switch (number) {
18                 case 1:
19                     dragon.moveRight();
20                     break;
21                 case 2:
22                     dragon.moveLeft();
23                     break;
24                 case 3:
25                     dragon.moveUp();
26                     break;
27                 case 4:
28                     dragon.moveDown();
29                     break;
30                 default:
31                     System.exit(0);
32                     break;
33             }
34             dragon.printPosition();
35         }
36     }
37 }
38

```

Move dragon

1. Move right
2. Move left
3. Move up
4. Move down
0. Exit

Enter number : 1

x : 1
y : 0

Move dragon

1. Move right
2. Move left
3. Move up
4. Move down
0. Exit

Enter number : 2

x : 0
y : 0

Move dragon

1. Move right
2. Move left
3. Move up
4. Move down
0. Exit

Enter number : 3

x : 0
y : 0

Move dragon

1. Move right
2. Move left
3. Move up
4. Move down
0. Exit

Enter number : 4

x : 0
y : 1

Move dragon

1. Move right
2. Move left
3. Move up
4. Move down
0. Exit

Enter number : 5

d:\DASPRO-JAVA>