



Nama

Raudhil Firdaus Naufal

NIM

2341720164

Kelas

1E

Jurusan

Teknologi Informasi

Prodi

Teknik Informatika

Percobaan 1: Deklarasi Class, Atribut dan Method.

```
Error: Could not find or load main class Jobsheet_2.bukuMain
Caused by: java.lang.ClassNotFoundException: Jobsheet_2.bukuMain
```

Pertanyaan:

1. Sebutkan dua karakteristik class atau object.

Class

- Class tidak akan ada jika belum dibuat objectnya.
- Memiliki proses pembuatan project dari suatu kelas yang disebut instansiasi

Object

- Mempunyai sesuatu.
- Melakukan sesuatu.

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

Atribut yang dimiliki oleh class Buku adalah 5.

- Judul
- Pengarang
- Halaman
- Stok
- Harga

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

Method yang dimiliki oleh class tersebut(buku22) adalah 4.

- tampilInformasi
- terjual
- restock
- 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;  
    } else{  
        System.out.println(x:"Stock tidak  
        tersedia.");  
    }  
}
```

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

Agar saat memanggil fungsi bisa menambahkan stock, jadi di dalam parameter dalam fungsi restock kita isi dengan angka yang akan masuk ke dalam int jumlah yang nanti akan ditambahkan dengan stok.

6. Commit dan push program ke GitHub.

Percobaan 2:Instansiasi Object, serta Mengakses Atribut dan Method.

```
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
```

Pertanyaan:

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

Baris kode program yang digunakan untuk instansiasi:

```
buku22 buku1 = new buku22();
```

Objek yang dihasilkan adalah buku1.

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

Cara mengakses atribut dan method dari suatu objek adalah dengan menambahkan .atributYangAkanDiakses setelah nama objek, setelah itu memberikan value atau nilainya. Tetapi sebelum itu harus melakukan instansiasi terlebih dahulu.

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

Karena pada method tampilInformasi() pertama belum melakukan perubahan atribut dengan menggunakan method yang lain, sedangkan pada tampilInformasi() yang kedua telah melakukan perubahan atribut stok dan harga dengan menggunakan method terjual() dan gantiHarga().

Percobaan 3: Membuat Konstruktor

```
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
```

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?

```
buku22 buku2 = new buku22(jud:"Self  
Reward",pg:"Maheera Ayesha",hal:160,  
stok:29,har:59000);
```

Yang dilakukan pada baris program tersebut adalah memasukkan nilai atribut pada parameter ke dalam method yang dimiliki pada class buku22.java.

3. Hapus konstruktor default pada class Buku, kemudian compile dan run program.

Bagaimana hasilnya? Jelaskan mengapa hasilnya demikian!

Hasilnya adalah error. Karena pada class bukuMain terdapat instansiasi yang memerlukan konstruktor default sehingga jika konstruktor default dihapus maka akan terjadi error.

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

Tidak, jika memang ingin diakses secara berurutan maka method bisa dipanggil secara berurutan, tetapi tanpa berurutan pun juga bisa, sesuai kebutuhan kode program.

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

```
buku22 bukuRaul = new buku22(jud:"Si  
Belalang",pg:"Raudhil",hal:110,stok:5,  
har:2000000);
```

6. Commit dan Push kode Program ke GitHub.

Latihan Praktikum.

1. Pada class buku yang telah dibuat tambahkan 3 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

```
void hitungHargaTotal(int hrg, int trjl){  
    hargaTotal = harga * trjl;  
}
```

- 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

```
void hitungDiskon(){  
    if (hargaTotal > 150000) {  
        System.out.println(x:"Selamat anda mendapatkan diskon sebesar 12%");  
        diskon = 0.12;  
    } else if (hargaTotal >= 75000 && hargaTotal <= 150000) {  
        System.out.println(x:"Selamat anda mendapatkan diskon sebesar 5%");  
        diskon = 0.5;  
    } else{  
        System.out.println(x:"Anda tidak mendapatkan diskon.");  
    }  
}
```

- Method `hitungHargaBayar()` digunakan untuk menghitung harga total setelah dikurangi diskon.

```
void hitungHargaBayar(){
    totalDiskon = hargaTotal * diskon;
    System.out.println("Diskon yang didapat: " + totalDiskon);
    hargaTotal -= totalDiskon;
    System.out.println("Total Harga yang didapat: " + hargaTotal);
}
```

```
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 8
Harga: Rp 60000
Diskon yang didapat: 36000.0
Total Harga yang didapat: 264000
```

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

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.

dragonMain

```
package Jobsheet_2;

import java.util.Scanner;

public class dragonMain {
    public static void main(String[] args) {

        String perintah = "";
        Scanner sc = new Scanner(System.in);

        dragon move = new dragon();

        move.x = 1;
        move.y = 1;
```



```

        System.out.println("Posisi x: " + move.x);
        System.out.println("Posisi y: " + move.y);

        while (!perintah.equalsIgnoreCase("atas") ||
!perintah.equalsIgnoreCase("bawah") || !perintah.equalsIgnoreCase("kanan") ||
!perintah.equalsIgnoreCase("kiri")) {

            System.out.print("Masukkan perintah(atas/bawah/kanan/kiri): ");
            perintah = sc.next();

            if (perintah.equalsIgnoreCase("atas")) {
                move.moveUp();
            } else if (perintah.equalsIgnoreCase("bawah")) {
                move.moveDown();
            } else if (perintah.equalsIgnoreCase("kanan")){
                move.moveRight();
            } else if (perintah.equalsIgnoreCase("kiri")){
                move.moveLeft();
            } else{
                System.out.println("Tolong Masukkan perintah yang tepat.");
            }

        }

    }

}

```

dragon

```

package Jobsheet_2;

public class dragon {

    int x,y,width = 10,height = 10;
    String perintah = "Over";

    void moveLeft(){
        x -= 1;
        System.out.println("Posisi x: " + x);
        System.out.println("Posisi y: " + y);
        detectCollision(x,y);
    }

}

```

```

void moveRight(){
    x += 1;
    System.out.println("Posisi x: " + x);
    System.out.println("Posisi y: " + y);
    detectCollision(x,y);
}

void moveUp(){
    y -= 1;
    System.out.println("Posisi x: " + x);
    System.out.println("Posisi y: " + y);
    detectCollision(x,y);
}

void moveDown(){
    y += 1;
    System.out.println("Posisi x: " + x);
    System.out.println("Posisi y: " + y);
    detectCollision(x,y);
}

void detectCollision(int x, int y){
    if (x > 10 || x < 1 || y > 10 || y < 1) {
        System.out.println("Game Over!");
        System.exit(0);
    }
}
}

```

Output.

```

Masukkan perintah(atas/bawah/kanan/kiri): kanan
Posisi x: 2
Posisi y: 3
Masukkan perintah(atas/bawah/kanan/kiri): kanan
Posisi x: 3
Posisi y: 3
Masukkan perintah(atas/bawah/kanan/kiri): kanan
Posisi x: 4
Posisi y: 3
Masukkan perintah(atas/bawah/kanan/kiri): atas
Posisi x: 4
Posisi y: 2
Masukkan perintah(atas/bawah/kanan/kiri): atas
Posisi x: 4
Posisi y: 1
Masukkan perintah(atas/bawah/kanan/kiri): atas
Posisi x: 4
Posisi y: 0
Game Over!

```