

MODUL 2

ENKAPSULASI (ENCAPSULATION)

1.1 TUJUAN PRAKTIKUM

Mahasiswa dapat memahami konsep enkapsulasi dalam pemrograman berorientasi objek dan menggunakannya dengan baik.

1.2 MATERI

- Enkapsulasi adalah mekanisme dalam PBO untuk mencegah/mengijinkan property atau method diakses dari luar Class. Kenapa harus dilindungi? Tujuannya adalah agar Class tersebut dapat berjalan sebagaimana mestinya sesuai dengan apa yang diinginkan oleh programmer. Untuk melakukan enkapsulasi digunakan kata kunci / *access modifier* sebagai berikut:
 1. **public** : bisa diakses oleh class manapun / luar class, bisa diakses oleh anggota lain dalam class, bisa diakses oleh anggota dari class turunan
 2. **protected** : tidak bisa diakses dari luar class, bisa diakses anggota lain dalam class, bisa diakses oleh anggota dari class turunan
 3. **private** : tidak dapat diakses dari luar class, bisa diakses anggota lain dalam class, tidak bisa diakses anggota dari class turunan
- Di Java Anda bisa memberikan modifier diatas di depan kata kunci **class** berhubungan dengan akses class dalam package, tapi di bahasa pemrograman lain tidak ada
- Terdapat beberapa hal yang harus Anda perhatikan dalam menentukan hak akses dari property atau method yaitu:

1. Usahakan semua property dan method **PRIVATE**, jika akan digunakan oleh Class turunan baru diubah ke **PROTECTED**, dan gunakan **PUBLIC** seminimal mungkin.
2. Jika Anda lupa memberi kata kunci modifier maka Java akan menganggap property/method tersebut adalah **PUBLIC**.
3. Karena semua property umumnya private maka biasanya perlu dibuat method **GETTER** dan **SETTER**. Getter digunakan untuk mendapatkan nilai dari property dan Setter digunakan untuk mengubah nilai property.

1.3 PRAKTIKUM

- Buat Class baru dengan mengklik kanan icon **Java Source Package** di jendela *Project* lalu pilih **New -> Java Class**. Di bagian Class Name diisi dengan nama **BangunDatar** dan kemudian tekan tombol **Finish**.
- Selanjutnya ketikkan kode berikut di Class Bangun Datar:

```
public class BangunDatar {  
    // property panjang dan lebar  
    private double panjang;  
    private double lebar;  
  
    // getter dan setter panjang  
    public double getPanjang() {  
        return panjang;  
    }  
    public void setPanjang(double panjang) {  
        // setter dengan validasi  
        if (panjang < 1) {  
            this.panjang = 1;  
        } else {  
            this.panjang = panjang;  
        }  
    }  
    // getter dan setter lebar  
    public double getLebar() {  
        return lebar;  
    }  
    public void setLebar(double lebar) {  
        // ternary operator  
        this.lebar = lebar < 1 ? 1 : lebar;  
    }  
  
    // method getLuas  
    public double getLuas() {  
        return panjang * lebar;  
    }  
}
```

- Untuk beberapa contoh menggunakan class diatas, ketikkan kode berikut di file **Main.java** (class **Main**):

```

package your.app.package;

public class Main {

    public static void main(String[] args) {
        BangunDatar bd1 = new BangunDatar();
        System.out.println(bd1.getPanjang()); // dapatkan nilai property
        bd1.setPanjang(10); // mengeset nilai property
        bd1.setLebar(3);
        System.out.println(bd1.getLuas());
    }
}

```

- Jalankan aplikasi dan Anda akan dapat melihat hasil penggunaan class BangunDatar yang telah kita buat

1.4 TUGAS

Selesaikan permasalahan berikut ini dengan menggunakan class dan objek!

1. Buat sebuah Class Bus dengan private property jumlah penumpang dan kapasitas
2. Buat setter untuk mengeset nilai kapasitas bus
3. Buat getter untuk mengetahui jumlah penumpang saat ini dan kapasitasnya
4. Buat setter untuk menambah dan mengurangi penumpang saat ini dan validasi input agar tidak lebih dari kapasitas bus atau lebih kecil dari 0
5. Buat bus yang lebih besar dari bus sebelumnya dengan membuat objek baru