

# Pemrograman Berorientasi Objek

*Tugas Kelas Abstrak & Interface*



Dipersiapkan Oleh :

< Chandra Harkat Raharja> - < 233040089>

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PASUNDAN**

**BANDUNG**

**2024**

Link Github: [Pemrograman-Berbasis-Objek\\_233040089/Pertemuan11](https://github.com/ComradeChandra/Pemrograman-Berbasis-Objek_233040089/Pertemuan11)  
at main · ComradeChandra/Pemrograman-Berbasis-Objek\_233040089

## -LATIHAN 1

Latihan ke-1 ini memberikan contoh program kelas abstrak. Kelas Bentuk adalah kelas abstrak yang salah satu methodnya abstrak. Method abstrak tersebut akan diimplementasikan oleh kelas Lingkaran dan kelas Tabung.

- Buatlah Kelas Bentuk Seperti di Bawah ini:

```
public abstract class Bentuk {  
    protected double PHI = 3.14;  
    protected int jari2;  
  
    public Bentuk(int jari2) {  
        super();  
        this.jari2 = jari2;  
    }  
  
    public abstract double luas();  
  
    // Setter & Getter  
  
    public int getJari2() {  
        return jari2;  
    }  
  
    public void setJari2(int jari2) {  
        this.jari2 = jari2;  
    }  
}
```

-Jawab:

```
Pertemuan11 > J Bentuk.java > ...  
1  package Pertemuan11;  
2  
3  public abstract class Bentuk {  
4      protected double PHI = 3.14;  
5      protected int jari2;  
6  
7      public Bentuk(int jari2) {  
8          super();  
9          this.jari2 = jari2;  
10     }  
11  
12     public abstract double luas();  
13  
14     // Setter & Getter  
15     public int getJari2() {  
16         return jari2;  
17     }  
18  
19     public void setJari2(int jari2) {  
20         this.jari2 = jari2;  
21     }  
22 }  
23
```

- Buatlah Kelas Lingkaran Seperti Di Bawah Ini:

```
public class Lingkaran extends Bentuk {  
  
    public Lingkaran(int jari2) {  
        super(jari2);  
    }  
  
    /*  
     * L:  $\text{PHI} * r * r$   
     */  
  
    @Override  
    public double luas() {  
        return PHI*jari2*jari2;  
    }  
  
}
```

-Jawab:

```
Pertemuan11 > J Lingkaran.java > Lingkaran > luas()  
1 package Pertemuan11;  
2  
3 public class Lingkaran extends Bentuk {  
4  
5     public Lingkaran(int jari2) {  
6         super(jari2);  
7     }  
8  
9     /*  
10     * L:  $\text{PHI} * r * r$   
11     */  
12     @Override  
13     public double luas() {  
14         return PHI * jari2 * jari2;  
15     }  
16 }  
17
```

-Buatlah Kelas Tabung Seperti Di bawah Ini:

```
public class Tabung extends Bentuk {  
    private int tinggi;  
  
    public Tabung() {  
        super(0);  
    }  
  
    public Tabung(int jari2, int tinggi) {  
        super(jari2);  
        this.tinggi = tinggi;  
    }  
  
    /*  
     * L:  $2\pi \cdot r \cdot (r+t)$   
     */  
  
    @Override  
    public double luas()  
    {  
        return 2*PHI*jari2*(jari2+tinggi);  
    }  
  
    // Setter & Getter  
  
    public int getTinggi() {  
        return tinggi;  
    }  
  
    public void setTinggi(int tinggi) {  
        this.tinggi = tinggi;  
    }  
  
}
```

-Jawab:

```
Pertemuan11 > J Tabung.java > ...  
1 package Pertemuan11;  
2  
3 public class Tabung extends Bentuk {  
4     private int tinggi;  
5  
6     public Tabung() {  
7         super(jari2:0);  
8     }  
9  
10    public Tabung(int jari2, int tinggi) {  
11        super(jari2);  
12        this.tinggi = tinggi;  
13    }  
14  
15    /*  
16    * L:  $2\pi \cdot r \cdot (r + t)$   
17    */  
18    @Override  
19    public double luas() {  
20        return 2 * PHI * jari2 * (jari2 + tinggi);  
21    }  
22  
23    // Setter & Getter  
24    public int getTinggi() {  
25        return tinggi;  
26    }  
27  
28    public void setTinggi(int tinggi) {  
29        this.tinggi = tinggi;  
30    }  
31 }  
32
```

## PERTANYAAN:

1. Sebutkan method abstrak yang terdapat di kelas abstrak Bentuk !
2. Jelaskan apa perbedaannya method luas di kelas Bentuk dan di kelas Lingkaran dan kelas Tabung!
3. Jelaskan bagaimana jika method luas di kelas Tabung dihilangkan atau tidak mengimplementasikannya!
4. Buatlah kelas main sehingga outputnya seperti dibawah ini:

```
== Lingkaran ==  
jari2: 10  
luas:314.0  
== Tabung ==  
jari2: 10, Tinggi:5  
luas:942.0000000000001
```

5. Buatlah diagram kelas dari contoh program kelas abstrak!

## JAWABAN:

1. Di dalam kelas Bentuk, terdapat satu method abstrak yaitu:

```
public abstract double luas();
```

Method ini hanya dideklarasikan tanpa isi atau implementasi, dan harus diimplementasikan oleh semua kelas yang mewarisi Bentuk.

2. Di kelas Bentuk, method luas() hanya didefinisikan sebagai method abstrak. Artinya, method ini **tidak memiliki isi** dan hanya berfungsi sebagai "kontrak" bagi kelas turunan.

- 2.a. Di kelas Lingkaran, method luas() di-*override* dan berisi rumus luas lingkaran, yaitu:

```
PHI × jari-jari × jari-jari
```

- 2.b. Di kelas Tabung, method luas() juga di-*override* dan berisi rumus luas permukaan tabung, yaitu:

```
2 × PHI × jari-jari × (jari-jari + tinggi)
```

Jadi perbedaannya adalah:

- Di Bentuk, method ini hanya berupa deklarasi.

- Di Lingkaran dan Tabung, method ini memiliki isi yang sesuai dengan bentuknya masing-masing.

3. Jika method `luas()` tidak diimplementasikan di kelas Tabung, maka akan muncul error saat kompilasi. Hal ini karena Tabung adalah turunan dari Bentuk, dan kelas Bentuk memiliki method abstrak `luas()`.

4. Jika method `luas()` tidak diimplementasikan di kelas Tabung, maka akan muncul **error saat kompilasi**. Hal ini karena Tabung adalah turunan dari Bentuk, dan kelas Bentuk memiliki method abstrak `luas()`.

Dalam Java, setiap kelas turunan dari kelas abstrak **wajib mengimplementasikan semua method abstrak**. Jika tidak, maka kelas turunan tersebut juga harus ditandai sebagai abstract. Karena Tabung tidak ditandai sebagai abstract, maka method `luas()` wajib ditulis.

## LATIHAN-2

Latihan ke-2 ini memberikan contoh program interface. Interface Kartu mempunyai semua method abstrak yaitu otentikasi dan encode. Interface Kartu harus implementasikan oleh kelas konkret yaitu kelas KartuElektronik. Kartu elektronik menggunakan interface Kartu menggunakan keyword implements.

- Buatlah interface Kartu seperti dibawah ini

```
public interface Kartu
{
    public boolean otentikasi(String pin);
    public String encode(String pin);
}
```

Jawab:

```
Pertemuan11 > J Kartu.java > ...
1  package Pertemuan11;
2
3  public interface Kartu
4  {
5      public boolean otentikasi(String pin);
6      public String encode(String pin);
7      → public String decode(String encodedPin);
8  }
```

-Buatlah kelas KartuElektronik seperti dibawah ini

```
public class KartuElektronik implements Kartu {
    private String kodeBank;
    private String pin;

    public KartuElektronik(String kodeBank, String pin) {
        super();
        this.kodeBank = kodeBank;
        this.pin = pin;
    }

    @Override
    public boolean otentikasi(String pinInput) {
        if (pin.equals(pinInput))
            return true;
        else
            return false;
    }

    @Override
    public String encode(String pin) {
        // melakukan enkripsi inputan PIN
        return null;
    }
}
```

-Jawab

```
Pertemuan11 > J KartuElektronik.java > ...
1  package Pertemuan11;
2
3  public class KartuElektronik implements Kartu {
4      private String kodeBank;    The value of the field KartuElektronik
5      private String pin;
6
7      public KartuElektronik(String kodeBank, String pin) {
8          super();
9          this.kodeBank = kodeBank;
10         this.pin = pin;
11     }
12
13     @Override
14     public boolean otentikasi(String pinInput) {
15         if (pin.equals(pinInput))
16             return true;
17         else
18             return false;
19     }
20
21     @Override
22     public String encode(String pin) {
23         // melakukan enkripsi inputan PIN
24         return null;
25     }
26
27 }
```

-Buatlah kelas InterfaceMain seperti dibawah ini

```
public class InterfaceMain {

    public static void main(String[] args) {
        KartuElektronik kartu = new KartuElektronik("IF111", "123");
        System.out.println("Otentikasi:" + kartu.otentikasi("123"));
    }

}
```

-Jawab:

```
Pertemuan11 > J InterfaceMain.java > ...
1  package Pertemuan11;
2
3  public class InterfaceMain {
4      Run | Debug
5      public static void main(String[] args) {
6          KartuElektronik kartu = new KartuElektronik(kodeBank:"IF111", pin:"123");
7          System.out.println("Otentikasi: " + kartu.otentikasi(pinInput:"123"));
8      }
9  }
```

## -Tugas

Buatlah contoh kasus lainnya program kelas abstrak dan interface seperti latihan yang sudah anda kerjakan dalam Bahasa pemrograman Java. Laporan yang dikumpulkan adalah rancangan kelas diagram dan sintaks kode programnya serta outputnya dari kode program tersebut.

Jawab:

Judul kasus: Sistem Autentikasi Pengguna di Aplikasi Digital

Penjelasan:

Pada program ini, dibuat sebuah simulasi sederhana mengenai sistem otentikasi pengguna yang terdiri dari dua jenis akun: Admin dan User Biasa. Setiap pengguna memiliki username dan password, serta harus bisa melakukan proses otentikasi dan enkripsi data.

Untuk menerapkan konsep abstraksi, digunakan sebuah kelas abstrak bernama Pengguna yang memuat struktur dasar pengguna dan method abstrak autentikasi(). Lalu, digunakan juga sebuah interface bernama Enkripsi yang berisi deklarasi method encode() untuk proses enkripsi data.

Dua kelas konkret yaitu Admin dan UserBiasa kemudian mewarisi dari Pengguna dan mengimplementasikan interface Enkripsi.

### Penjelasan Struktur Program

#### 1. Kelas abstrak Pengguna

Merupakan kelas dasar yang menyimpan atribut username dan password. Di dalamnya terdapat method abstrak autentikasi() yang harus diimplementasikan oleh kelas turunannya.

```
Pertemuan11 > TugasModul > J Pengguna.java > ...
1  package Pertemuan11.TugasModul;
2
3  public abstract class Pengguna {
4      protected String username;
5      protected String password;
6
7      public Pengguna(String username, String password) {
8          this.username = username;
9          this.password = password;
10     }
11
12     public abstract boolean autentikasi(String userInput, String passInput);
13 }
14
```

## 2. Interface Enkripsi

Interface ini mendefinisikan method encode() untuk melakukan enkripsi data. Tujuannya agar setiap kelas yang membutuhkan fitur enkripsi dapat mengimplementasikan interface ini sesuai kebutuhan masing-masing.

```
Pertemuan11 > TugasModul > J Enkripsi.java > ...
1  package Pertemuan11.TugasModul;
2  public interface Enkripsi {
3      String encode(String data);
4  }
5
```

## 3. Kelas Admin

Kelas ini mewarisi dari Pengguna dan mengimplementasikan Enkripsi. Metode autentikasi() akan memverifikasi username dan password, sedangkan encode() akan membalik string (simulasi enkripsi).

```
Pertemuan11 > TugasModul > J Admin.java > ...
1  package Pertemuan11.TugasModul;
2
3  public class Admin extends Pengguna implements Enkripsi {
4
5      public Admin(String username, String password) {
6          super(username, password);
7      }
8
9      @Override
10     public boolean autentikasi(String userInput, String passInput) {
11         return username.equals(userInput) && password.equals(passInput);
12     }
13
14     @Override
15     public String encode(String data) {
16         return "ADMIN-" + new StringBuilder(data).reverse().toString();
17     }
18 }
19
```



#### 4. Kelas UserBiasa

Mirip seperti Admin, namun metode encode() di kelas ini akan menyamarkan karakter string dengan tanda \* sebagai bentuk penyembunyian informasi.

```
Pertemuan11 > TugasModul > J UserBiasa.java > ...
1  package Pertemuan11.TugasModul;
2
3  public class UserBiasa extends Pengguna implements Enkripsi {
4
5      public UserBiasa(String username, String password) {
6          super(username, password);
7      }
8
9      @Override
10     public boolean autentikasi(String userInput, String passInput) {
11         return username.equals(userInput) && password.equals(passInput);
12     }
13
14     @Override
15     public String encode(String data) {
16         return "USER-" + data.replaceAll(".", "*");
17     }
18 }
19
```

#### 5. Kelas Main

Berfungsi sebagai kelas utama untuk mengeksekusi program. Di sini dibuat objek Admin dan UserBiasa, lalu ditampilkan hasil dari proses otentikasi dan enkripsi datanya.

```
Pertemuan11 > TugasModul > J Main.java > ...
1  package Pertemuan11.TugasModul;
2  public class Main {
3      Run | Debug
4      public static void main(String[] args) {
5          Pengguna admin = new Admin(username:"admin123", password:"passadmin");
6          Pengguna user = new UserBiasa(username:"user321", password:"passuser");
7
8          System.out.println("== Admin ==");
9          System.out.println("Autentikasi: " + admin.autentikasi(userInput:"admin123", passInput:"passadmin"));
10         System.out.println("Enkripsi: " + ((Enkripsi)admin).encode(data:"RahasiaAdmin"));
11
12         System.out.println("\n== User Biasa ==");
13         System.out.println("Autentikasi: " + user.autentikasi(userInput:"user321", passInput:"passuser"));
14         System.out.println("Enkripsi: " + ((Enkripsi)user).encode(data:"PrivasiUser"));
15     }
16 }

```

### Output:

```
== Admin ==
Autentikasi: true
Enkripsi: ADMIN-nimdAaisahaR

== User Biasa ==
Autentikasi: true
Enkripsi: USER-*****
PS D:\KuliahSMT4\Pemrograman Berbasis Objek (PBO)>
```

