

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
PERTEMUAN KE-4
INHERITANCE

TUJUAN PRAKTIKUM

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu memahami konsep pewarisan antar objek dan menerapkannya dalam program.

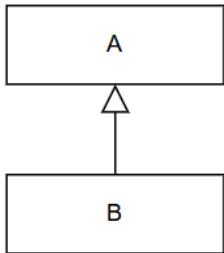
TOOLS

Tools yang diperlukan untuk melakukan praktikum ini adalah Java Development Kit (JDK) untuk mengkompilasi dan menjaalankan program Java serta code editor atau IDE untuk menulis program Java.

LANDASAN TEORI

Inheritance adalah usaha sebuah kelas untuk mewarisi atribut maupun metode dari kelas yang lain. Namun tidak semua atribut maupun metode bisa diwariskan kepada kelas yang lain. Access modifier pada elemen kelas (atribut dan method) menentukan mana elemen yang dapat diwariskan. Dalam Java terdapat 4 jenis access modifier, yaitu default, private (-), protected (#), dan public (+). Tabel 1 menunjukkan contoh mapping hubungan pewarisan dari class diagram ke dalam program Java.

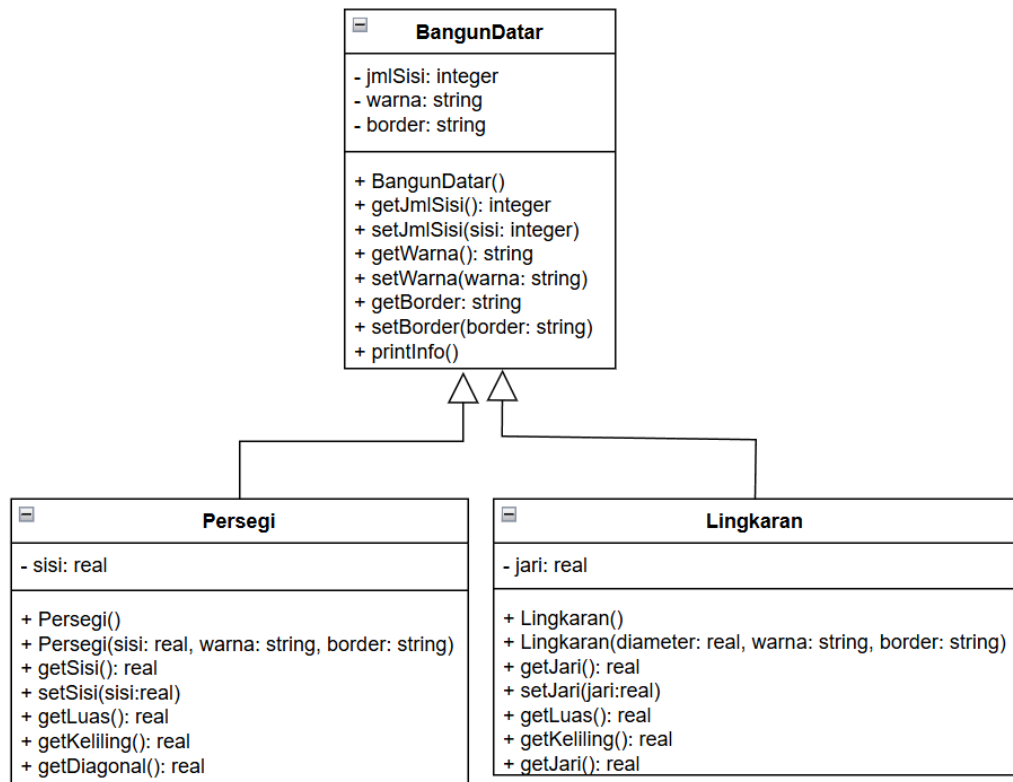
Tabel 1. Relasi Inheritance dan Implementasinya

Notasi UML	Implementasi Program Java
	<pre>class A { ... } class B extends A { ... }</pre>

LANGKAH PRAKTIKUM

Bagian 1 – Simple Inheritance

1. Tuliskan nama file, deskripsi, pembuat, dan tanggal pada bagian awal setiap file Anda sebagai komentar.
2. Implementasikan kelas BangunDatar, Persegi, dan Lingkaran ke dalam program Java dengan mengimplementasikan hubungan pewarisan, yaitu kelas Persegi dan Lingkaran mewarisi kelas BangunDatar, seperti pada Gambar 1.



Gambar 1. Class Diagram BangunDatar

Berikut adalah cuplikan kode untuk kelas BangunDatar:

```

12 public class BangunDatar {
13     private int jmlSisi;
14     private String warna;
15     private String border;
16
17     public BangunDatar () {
18     }
19
20     public int getJmlSisi () {
21         return jmlSisi;
22     }
23
24     public void setJmlSisi (int jmlSisi) {
25         this.jmlSisi = jmlSisi;
26     }
27
28     public String getWarna () {
29         return warna;
30     }
31
32     public void setWarna (String warna) {
33         this.warna = warna;
34     }
35 }
  
```

```

35 public String getBorder() {
36     return border;
37 }
38
39 public void setBorder(String border) {
40     this.border = border;
41 }
42
43 public void printInfo() {
44     System.out.println("Jumlah sisi: " + jmlSisi);
45     System.out.println("Warna: " + warna);
46     System.out.println("Border: " + border);
47 }
48 }

```

Berikut ini adalah cuplikan kode untuk kelas Persegi. Lengkapi kelas tersebut dengan method lainnya seperti yang ada pada Gambar 1.

```

11 public class Persegi extends BangunDatar {
12     private double sisi;
13
14     public Persegi() {
15         setJmlSisi(4);
16     }
17
18     public Persegi(double sisi, String warna, String border) {
19         this.sisi = sisi;
20         setWarna(warna);
21         setBorder(border);
22         setJmlSisi(4);
23     }
24
25     public double getSisi() {
26         return sisi;
27     }
28
29     public void setSisi(double sisi) {
30         this.sisi = sisi;
31     }
32 }

```

Apakah konstruktor berparameter pada kelas Persegi dapat direalisasikan seperti berikut? Mengapa?

```

public Persegi(double sisi, String warna, String border) {
    this.jmlSisi = 4;
    this.warna = warna;
    this.border = border;
    this.sisi = sisi;
}

```

3. Buatlah sebuah main class dan main method dalam kelas tersebut. Dalam main method tersebut buatlah beberapa contoh objek dari class Persegi dan Lingkaran, kemudian coba panggil method-method yang telah dibuat dalam class Persegi dan Lingkaran tersebut.

Bagian 2 – Keyword super

Keyword super merujuk pada superclass dan dapat digunakan untuk memanggil konstruktor method pada superclass.

1. Pada kelas BangunDatar tambahkan sebuah konstruktor dengan parameter seperti berikut:

```
19 public BangunDatar(int jmlSisi, String warna, String border){
20     this.jmlSisi = jmlSisi;
21     this.warna = warna;
22     this.border = border;
23 }
```

2. Modifikasilah isi konstruktor berparameter pada kelas Persegi maupun Lingkaran dengan memanfaatkan keyword super. Berikut adalah contoh pada kelas Persegi:

```
18 public Persegi(double sisi, String warna, String border){
19     super(4, warna, border);
20     this.sisi = sisi;
21 }
```

Bagian 3 – Overriding Method

Overriding method digunakan untuk mendefinisikan ulang method yang ada di superclass, sehingga memiliki body yang berbeda dengan superclass tetapi tetap memiliki signature yang sama.

1. Overriding method printInfo() pada class Persegi dan Lingkaran, sehingga printInfo() pada masing-masing kelas tersebut menampilkan informasi nilai atributnya. Berikut adalah contoh overriding method printInfo() pada kelas Persegi:

```
31 @Override
32 public void printInfo() {
33     System.out.println("Jumlah sisi: " + getJmlSisi());
34     System.out.println("Warna: " + getWarna());
35     System.out.println("Border: " + getBorder());
36     System.out.println("Sisi: " + sisi);
37 }
```

2. Keyword super juga dapat digunakan untuk memanggil versi asli method pada superclass. Berikut adalah contoh pemanggilan method printInfo() milik kelas BangunDatar di dalam kelas Persegi:

```
41 @Override
42 public void printInfo() {
43     super.printInfo();
44     System.out.println("Sisi: " + sisi);
45 }
```

Override pada Static Method

1. Tambahkan sebuah atribut static counterBangunDatar pada kelas BangunDatar, lalu modifikasilah konstruktornya, dan tambahkan method static printCounterBangunDatar(), seperti berikut:

```

12 public class BangunDatar {
13     private int jmlSisi;
14     private String warna;
15     private String border;
16     private static int counterBangunDatar = 0;
17
18     public BangunDatar () {
19         counterBangunDatar++;
20     }
21
22     public BangunDatar(int jmlSisi, String warna, String border){
23         this.jmlSisi = jmlSisi;
24         this.warna = warna;
25         this.border = border;
26         counterBangunDatar++;
27     }
28
29     public static void printCounterBangunDatar() {
30         System.out.println("Jumlah Objek Bangun Datar: " + counterBangunDatar);
31     }
32 }

```

2. Coba lakukan overriding pada method printCounterBangunDatar(). Bagaimana hasilnya?

Bagian 4 – Access Modifier Protected

Lingkup Akses protected dalam Java adalah sebagai berikut:

1. Dapat diakses dalam kelas yang sama
2. Dapat diakses oleh subclass (pewarisan/inheritance), bahkan jika subclass berada di paket yang berbeda
3. Dapat diakses oleh kelas lain dalam paket yang sama
4. Tidak dapat diakses oleh kelas di luar paket jika bukan subclass

Modifikasilah atribut class BangunDatar dengan access modifier protected. Setelah modifikasi tersebut, apakah konstruktor berparameter pada kelas Persegi dapat direalisasikan seperti berikut? Mengapa?

```

public Persegi(double sisi, String warna, String border){
    this.jmlSisi = 4;
    this.warna = warna;
    this.border = border;
    this.sisi = sisi;
}

```

Bagian 5 – Keyword Final

Keyword final yang diberikan pada kelas menjadikan kelas tersebut tidak dapat diwariskan, sedangkan keyword final yang diberikan pada method menjadikan method tersebut tidak dapat di-override pada subkelasnya.

Lakukan eksperimen penggunaan keyword final seperti berikut:

1. Tambahkan keyword final pada kelas BangunDatar. Bagaimana pengaruhnya terhadap kelas Persegi dan Lingkaran?
2. Hapus kembali keyword final pada kelas BangunDatar, lalu tambahkan keyword final pada method printInfo() di dalam kelas BangunDatar. Bagaimana pengaruhnya dengan method printInfo() pada kelas Persegi dan Lingkaran?

LATIHAN

Sebuah perguruan tinggi memiliki pegawai yang terdiri atas dosen dan tenaga kependidikan (tendik). Setiap pegawai memiliki NIP (Nomor Induk Pegawai), nama, tanggal lahir, Terhitung Mulai Tanggal (TMT) bekerja, dan gaji pokok. Dosen terdiri atas 2 jenis, yaitu dosen tetap dan dosen tamu. Dosen tetap memiliki identitas NIDN (Nomor Induk Dosen Nasional), sedangkan dosen tamu memiliki identitas NIDK (Nomor Induk Dosen Khusus). Dosen tetap memiliki Bata Usia Pensiun (BUP) 65 tahun, dan mendapat tunjangan $2\% \times$ masa kerja (dalam tahun). Dosen tamu memiliki tanggal berakhir kontrak dan mendapat tunjangan $2,5\% \times$ gaji pokok. Tendik memiliki BUP 55 tahun dan mendapatkan tunjangan $1\% \times$ masa kerja (tahun). Masa kerja dihitung dari TMT hingga tanggal saat ini. BUP jatuh pada tanggal 1 bulan berikutnya dari tanggal lahir ditambah usia BUP. Tendik bekerja pada salah satu bidang, yaitu Akademik, Kemahasiswaan, atau Sumber Daya. Dosen tetap atau dosen tamu bekerja pada fakultas tertentu.

Berdasarkan kondisi tersebut buatlah desain *class diagram* yang tepat dengan memanfaatkan relasi *inheritance*, kemudian implementasikan desain anda dalam program Java. Pada setiap jenis pegawai nantinya memiliki method `printInfo()` yang menampilkan detail data pegawai sesuai dengan jenisnya.

Contoh tampilan informasi detail pegawai untuk Dosen Tetap:

NIP	: 9545647548
NIDN	: 78647324
Nama	: Andi
Tanggal Lahir	: 5 Mei 1990
TMT	: 1 Januari 2015
Jabatan	: Dosen Tetap
Fakultas	: Fakultas Sains dan Matematika
Masa Kerja	: 10 tahun 2 bulan
BUP	: 1 Juni 2055
Gaji Pokok	: Rp 5.000.000,00
Tunjangan	: $2\% \times 10 \times \text{Rp } 5.000.000,00 = \text{Rp. } 1.000.000,00$

Keterangan contoh tersebut:

- Informasi tanggal ditampilkan dalam format <angka hari> <nama bulan> <angka tahun>, contoh: 5 Mei 1990.
- Masa kerja ditampilkan dalam ... tahun ... bulan.
- Masa kerja dihitung dari TMT, yaitu 1 Januari 2015 sampai tanggal saat ini, yaitu 10 Maret 2025.
- BUP Dosen tetap adalah 65 tahun, dihitung dari tanggal lahir, yaitu 5 Mei 1990, dan jatuh pada tanggal 1 bulan berikutnya.
- Pada jenis Dosen Tamu, BUP digantikan dengan masa kontrak berakhir (dalam bulan) dihitung dari tanggal sekarang sampai tanggal berakhir kontrak, sedangkan NIDN digantikan dengan NIDK.
- Pada jenis Tendik, informasi fakultas digantikan dengan bidang tempat bekerja.

PELAPORAN

Selama sesi praktikum, laporkan hasil praktikum pada link <http://tiny.cc/pbo25>.

Lengkapi semua file program yang harus dikerjakan dalam modul ini dan kumpulkan hasil akhirnya di kulon maksimal H+3 setelah pelaksanaan praktikum.

*****Selamat Mengerjakan dan Berlatih *****