Fondasi Pemrograman

Modul Praktikum

# Topik

Inheritance

# Tujuan

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

1. Membuat membuat class inhetirance
2. Membuat constructor suatu class yang merupakan turunan dari class lain

# Alat dan Bahan

* Komputer dengan **Java Development Kit (JDK)** dan aplikasi BlueJ sudah terinstall
* Informasi download dan instalasi JDK dapat diperoleh di; <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
* Informasi download dan instalasi JDK dapat diperoleh di;

<http://www.bluej.org>

# Rangkuman/Teori

## [Membuat class turunan]

Inheritance merupakan mekanisme mendasar pada code reuse. Inheritance memungkinkan kita untuk extend functionalitas dari suatu object. Dengan kata lain, kita bisa membuat tipe baru dengan menambahkan field dan method (properties) dari tipe original-nya, sebagai hirarki dari classes yang ada.

Untuk membuat suatu class turunan kita hanya perlu menambahkan kata extends seperti contoh penulisan berikut:

class Student **extends** Person {

 ….

}

Berdasarkan contoh tersebut, berarti kita membuat class namanya Students yang diturunkan dari class Person.

## [Membuat method constructor class turunan]

Seperti yang sudah kita ketahui sebelumnya, bahwa method constructor adalah method yang biasa digunakan untuk menginisialisasi nilai dari fields pada class. Constructor untuk class turunan selain menginisialisasi nilai field pada class-nya sendiri, juga menginisialisasi nilai field dari class induknya. Untuk menginisialisasi field pada class induknya dengan instruksi **super()** yang artinya sama dengan menjalankan method constructor induknya dari class turunan. Jumlah dan urutan parameter pada instruksi super mengikuti urutan dan jumlah parameter pada method constructor induk.

Misal potongan class induk sebagai berikut;

**class Person{**

**public Person( String n, int ag, String ad, String p ) {**

**name = n;**

**age = ag;**

**address = ad;**

**phone = p;**

**}**

**….**

**}**

Dan potongan class turunan sebagai berikut;

class Student **extends** Person {

 double gpa;

public Student( String n, int ag, String ad, String p, double g ) {

 **super( n, ag, ad, p );**

 gpa = g;

 }

}

Berdasarkan kedua contoh tersebut, class Students yang diturunkan dari class Person memiliki constructor; public Student( String **n**, int **ag**, String **ad**, String **p**, double **g** ). Seperti dapat kita lihat bahwa class Student hanya membuat satu field yaitu; gpa, namun pada constructor disediakan sebanyak 5 parameter yang nantinya akan mengisi field (1 parameter untuk mengisi field gpa dan 4 parameter lainnya untuk mengisi field pada class induk).

Untuk mengisi field gpa digunakan statement gpa=g, artinya niai gpa akan diisi dari parameter g. Sedangkan untuk mengisi field pada class induk kita harus melalui constructor class induk (karena field tersebut memiliki type *private*, sehingga kita tidak bisa mengakses langsung field pada class induk) untuk itu maka kita gunakan perintah super(). Parameter yang diisi pada perintah super() mengikuti urutan parameter constructor. Perhatikan constructor class induk dan instruksi super() berikut;



## [Menggunakan method class induk]

Method pada class induk yang dapat di akses dari class turunan adalah method yang memiliki tipe publlic. Untuk mengaksesnya kita cukup menuliskan nama method tersebut. Perhatikan potongan class berikut;

Class induk:

class Person{

 ….

 public String toString( ) {

 return getName( ) + " " + getAge( ) + " " +getAddress( ) + " "+ getPhoneNumber( );

 }

 public String getName( ) {

 return name;

 }

 public int getAge( ) {

 return age;

 }

 public String getAddress( ) {

 return address;

 }

 public String getPhoneNumber( ) {

 return phone;

 }

 ….

}

Class turunan:

class Student extends Person {

 ….

 public String toString( ) {

 return **getName( )** + " " + **getAge( )** + " " +**getAddress( )**+ " "+ **getPhoneNumber( )**+

 " " + getGPA( );

 }

 ….

}

# Praktek

Pada bagian ini, akan dibahas mengenai praktek yang akan dilakukan. Untuk mempermudah pencatatan hasil praktek dan penarikan kesimpulan terhadap hasil praktek tersebut, Anda dapat menggunakan form pada **Appendix 1.**

## Praktek 1: Membuat class Hello

|  |
| --- |
| Hello |
| String namaint usia |
| Hello()Hello(parNama, parUsia)void isiData(parNama, parUsia)String sayHello()String lihatData()int lihatUsia()String lihatNama()void ubahUsia(parUsia)void ubahNama(parNama) |

1. Buat class hello dengan spesifikasi seperti diagram class di atas.
2. Uji setiap method pada clas tersebut melalui command line.

## Praktek 2: Membuat turunan (Hello2)

|  |
| --- |
| Hello2 |
| String prodiString univ |
| Hello2()Hello2(parNama, parUsia, parProdi, parUniv)void isiDataHello2(parNama, parUsia, parProdi, parUniv)String toString()int lihatProdi()String lihatUniv()void ubahProdi(parProdi)void ubahUniv(parUniv) |



1. Buat class turunan (Hello2) dengan spesifikasi seperti diagram class di atas.
2. Pada class Hello2 tersebut tambahkan field String Prodi dan field String Univ (seperti yang tertulis pada class diagram Hello2).
3. Pada class Hello2 tersebut tambahkan method-method seperti yang tertulis pada class diagram Hello2.
4. Uji setiap method pada clas tersebut melalui command line.

# Analisa Hasil Praktek

Berdasarkan praktek yang telah Anda lakukan, buat kesimpulan:

# Tugas

# Appendix 1: Form Data Praktek

Nama : ……………………….

NIM : ……………………….

Tgl : ……………………….

|  |  |
| --- | --- |
| **Penggalan *Source code*** | **Keluaran / *Output*** |
|  |  |
|  |  |
|  |  |