MAKALAH, PERCOBAAN, LATIHAN, DAN TUGAS

MODUL PRAKTIKUM 5

Disusun sebagai salah satu tugas

mata kuliah PBO I



Patricia Joanne

140810160065

Dikumpulkan tanggal

26 September 2017

PROGRAM STUDI S-1 TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

2017

**Tugas Pendahuluan**

1. Apa yang dimaksud dengan inheritance?

2. Buatlah contoh kasus yang menerapkan konsep inheritance!

3. Adakah perbedaan cara mengakses member class milik parent dan member class milik sendiri? Jelaskan melalui contoh! (Silahkan memanfaatkan jawaban soal nomor 2)

4. Apa yang dimaksud dengan konsep single inheritance?

5. Apa yang dimaksud dengan konsep multi level inheritance?

6. Ada berapa modifier untuk pengontrolan akses? Jelaskan masing-masing!

7. Apakah kegunaan kata kunci super? Jelaskan!

8. Apakah yang dimaksud dengan konstruktor tidak diwariskan?

1. Pengertian dari **inheritance** adalah dimana sebuah objek dapat mempunyai objek turunan. Dalam hal pemrograman makan dapat diimpelementasikan dengan sebuah class yang mempunyai class turunan. Class yang mempunyai class turunan disebut sebagai Parent Class atau Base Class, sedangkan class yang menjadi turunan dari Parent Class disebut sebagai Subclass atau Child Class. Secara sederhana misal Hewan adalah parent class maka child class bisa berupa ayam, kucing, bebek dll. Suatu subclass dapat mewarisi sebagian atau keseluruhan sifat dari parent classnya, denga kata lain sebuah subclass hanya merupakan perluasan dari parent class.

2. --- Parent ---

public class Induk {  
    String hartaInduk;  
        
    protected void setHartaInduk(String h) {   
        this.hartaInduk = h;  
    }

--- Child ---

public class Anak extends Induk{ //extend menyatakan turunan dari induk  
    String hartaAnak; //field atau tipe datanya  
      
    protected void setHartaAnak(String h) {  
        this.hartaAnak = h;  
    }

3. Pengaksesan member yang ada di parent class dari subclassnya tidak jauh berbeda dengan pengaksesan member subclass itu sendiri. Contoh: Suatu parent class dapat tidak mewariskan sebagian member-nya kepada subclassnya. Sejauh mana suatu member dapat diwariskan ke class lain, ataupun suatu member dapat diakses dari class lain, sangat berhubungan dengan access control (kontrol pengaksesan).

Dalam inheritance, member kelas yang diwariskan hanyalah variabel dan method saja. Sedangkan konstruktor parent class tidak diwariskan ke subclass. Tetapi ketika suatu obyek anak dibuat dalam artian ketika konstruktor anak dijalankan maka konstruktor parent class dijalankan terlebih dahulu dan selanjutnya menyelesaikan konstruktor anak.

4. Konsep single inheritance hanya memperbolehkan suatu sublass mempunyai satu parent class. Dengan konsep single inheritance ini, masalah pewarisan akan dapat diamati dengan mudah.

5. Perbedaan konsep multi level inheritance dengan konsep single inheritance yaitu memperbolehkan suatu subclass mempunyai subclass lagi.

6. Modifier akses berguna untuk memberikan hak akses kepada user. Terdiri dari:

* DEFAULT: Class member hanya dapat digunakan hanya pada class di paket yang sama. Tanpa penulisan modifier akses.
* PUBLIC: Class member dapat diakses darimana saja, baik di dalam paket maupun di luar paket  
  PRIVATE: Class member hanya dapat diakses di dalam class itu sendiri atau class yang telah didefinisikan.
* PROTECTED: Class member dapat diakses di dalam package yang sama dan sub classnya. Subclass dapat berada di paket yang berbeda subclassnya akan mewarisi member variabel.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Note | Private | Default/package | Protected | Public |
| Same Class | v | v | v | v |
| Same Package |  | v | v | v |
| Different package (sub class) |  |  | v | v |
| Different package (non – sub class) |  |  |  | v |

7. Subclass juga dapat memanggil constructor secara eksplisit dari superclass terdekat. Hal ini   
dilakukan  dengan  pemanggil  constructor  super.  Pemanggil  constructor  super  dalam   
constructor  dari  subclass  akan  menghasilkan  eksekusi  dari  superclass  constructor  yang   
bersangkutan, berdasar dari argumen sebelumnya.

Ada beberapa hal yang harus diingat ketika menggunakan pemanggil constuktor super:

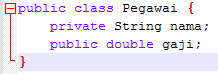
* Pemanggil super() harus dijadikan pernyataan pertama dalam constructor.
* Pemanggil super() hanya dapat digunakan dalam definisi constructor.
* Termasuk  constructor  this()  dan  pemanggil  super()  tidak  boleh  terjadi  dalam constructor yang sama.
* Pemakaian  lain  dari  super  adalah  untuk  menunjuk  anggota  dari  superclass (seperti reference this).

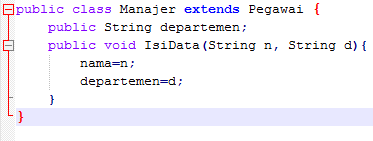
8. Dalam konsep pewarisan constructor kelas induk tidak akan diwariskan ke kelas induk. Untuk memfasilitasi pemanggilan fungsi constructor kelas induk oleh kelas anak maka Java menyediakan mekanisme rujukan ke kelas induk dengan menggunakan kata kunci super. Dengan menggunakan kata kunci ini, maka kita dapat mengakses semua variabel kelas induk walaupun variabel tersebut tidak diwariskan.

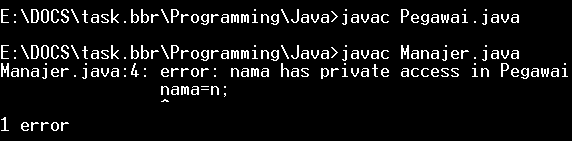
**Bab 5**

**Sifat Pewarisan**

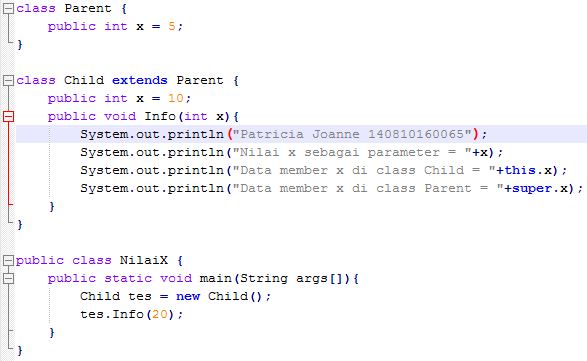
Percobaan 1: Penyembunyian informasi

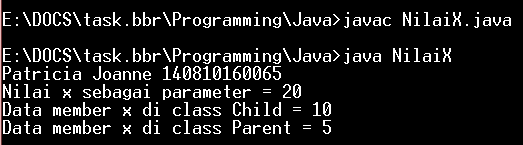




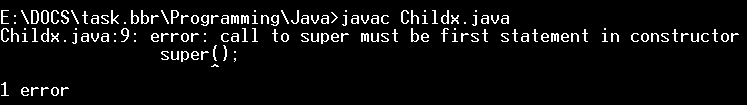
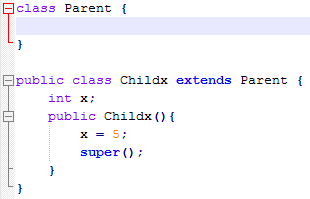


Percobaan 2: Menggunakan kata kunci super

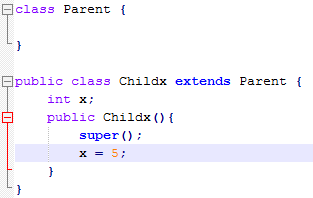


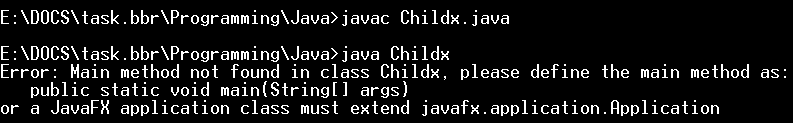


Percobaan 3: Konstruktor tidak diwariskan

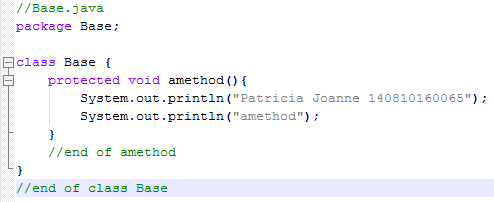


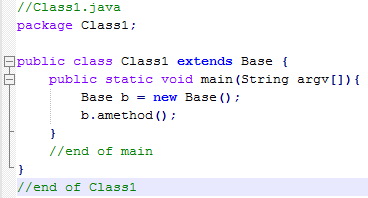
Seharusnya pemanggilan konstruktor untuk parent class harus dilakukan di baris pertama pada konstruktor subclassnya

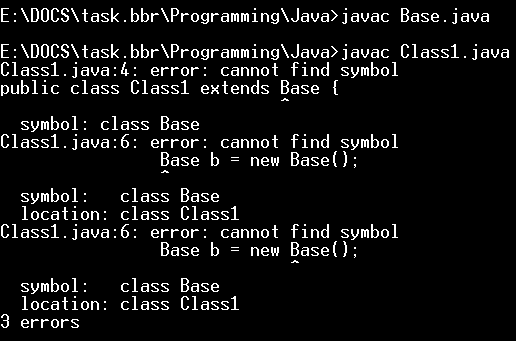




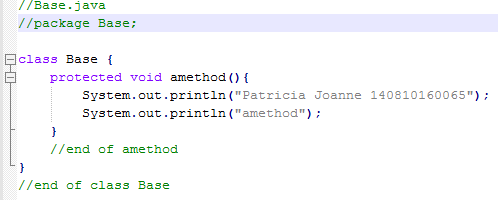
Latihan 1: Tempatkan class Base dan class Class1 di direktori yang sama. Apa yang terjadi ketika Class1.java dikompile dan dijalankan jika sebelumnya Base.java belum dikompile? Jelaskan!

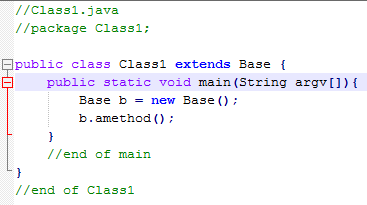






Pembuatan package menjadi penyebab error karena dalam pewarisan sifat dalam 1 direktori tidak perlu menggunakan package. Solusi:



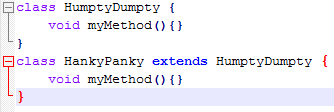




Latihan 2: Aturan overriding

a. Berdasarkan kode di bawah ini, akses modifier (public, protected atau private) apa yang diijinkan di tambahkan sebelum myMethod() baris 3?

b. Jika baris 3 seperti kode di bawah (apa adanya tanpa perubahan) keywords apa yang diijinkan ditambahkan sebelum myMethod baris 8?

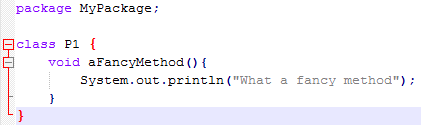
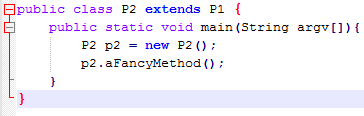


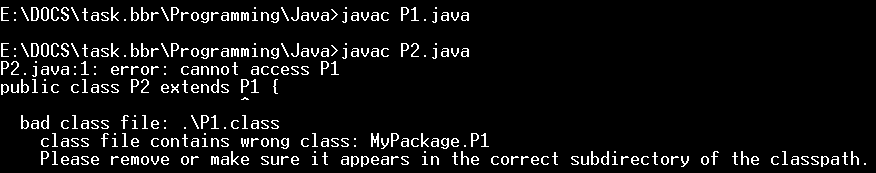
Berdasarkan Source code di atas, akses modifier yang diijinkan untuk ditambahkan sebelum myMethod() di baris 3 yaitu Private. Sedangkan, akses modifier yang diijinkan untuk ditambahkan sebelum myMethod() di baris 8 yaitu Public dan Protected.

Latihan 3:

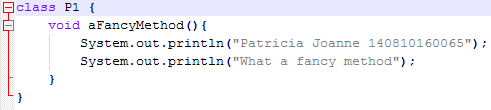
a. Apa yang terjadi bila kedua kode di bawah ini dikompile dan dijalankan dalam satu direktori? Jelaskan!

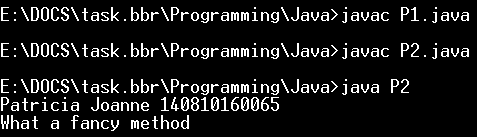
b. Bagaimana solusi supaya tidak terjadi error?

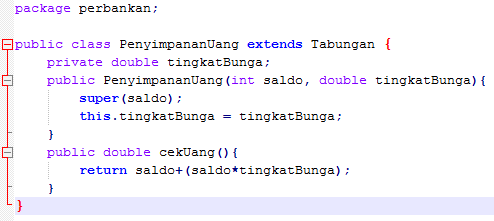


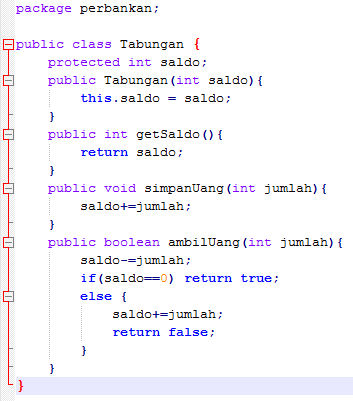
Seharusnya packagenya dihapus saja atau dibuat folder dengan nama MyPackage dan P1.java dimasukkan ke dalam folder tersebut. Jadi yang benar seperti ini.

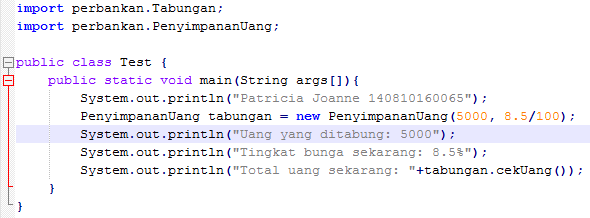


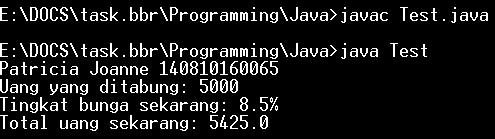


Latihan 4: Mengimplementasikan UML class diagram dalam program untuk package perbankan

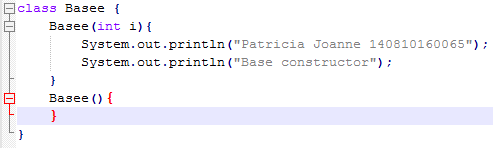


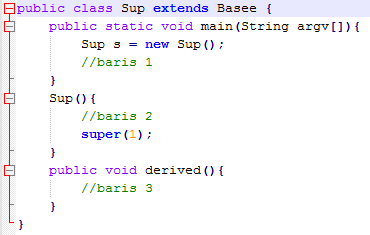


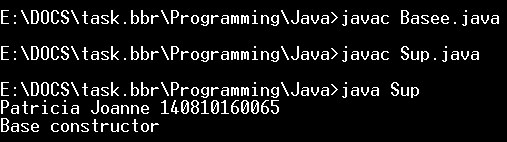




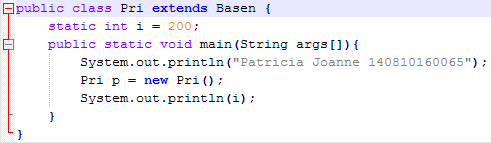
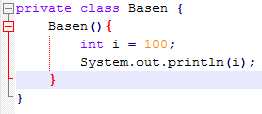
Latihan 5: Konstruktor tidak diwariskan





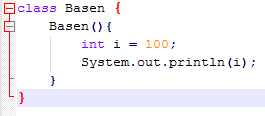


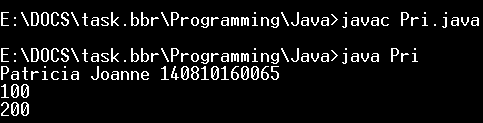
Latihan 6: Konstruktor tidak diwariskan



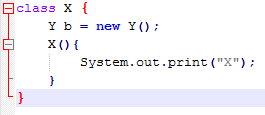
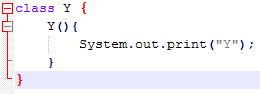


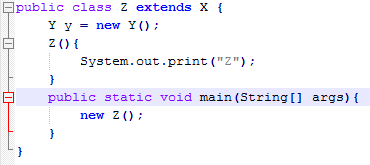
Class yang diprivate tidak bisa digunakan untuk file class lainnya. Seharusnya privatenya dihapus saja.

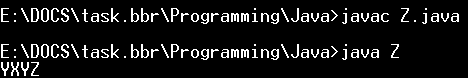




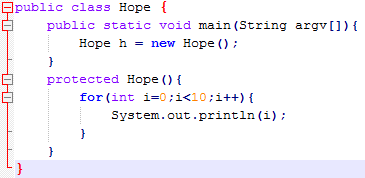
Latihan 7: Apa yang tampil di layar bila kode di bawah ini dijalankan?

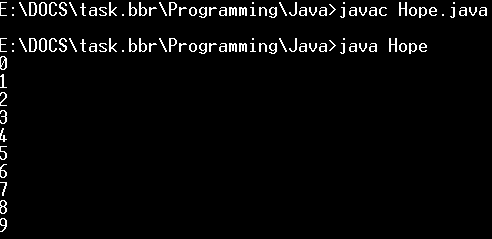




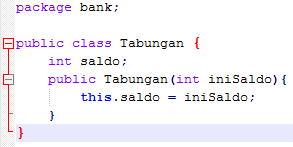


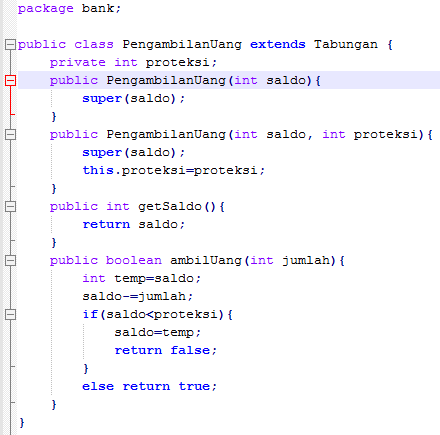
Latihan 8: Kompile dan jalankan program berikut! Apa yang terjadi? Jelaskan!

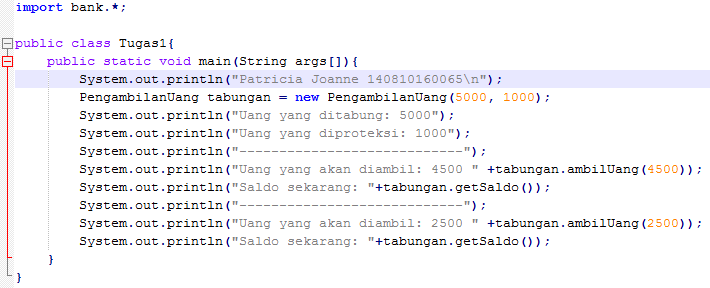


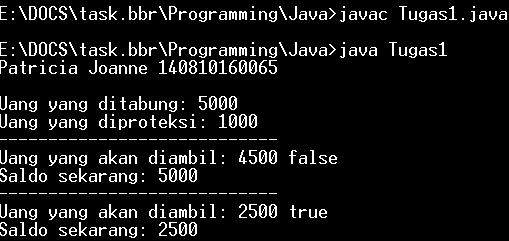


Tugas 1: Mengimplementasikan UML class diagram dalam program untuk package perbankan









Tugas 2: Mengimplementasikan UML class diagram dalam program

a. Buatlah kelas-kelas berdasarkan UML class diagram yang ada!



b. Selanjutnya buatlah kelas Tes.java yang membuat obyek-obyek serta mengeset nilai variabel seperti pada Tabel 5.2. dan tampilkan data per obyek.

