**JURUSAN TEKNIK INFORMATIKA**

**UNIVERSITAS SANATA DHARMA**

**PRAKTIKUM PBO I**

**MODUL 8 Pewarisan**

1. **TUJUAN PRAKTIKUM**
2. Mahasiswa mampu memahami konsep pewarisan.
3. Mahasiswa mampu merancang kelas yang mewarisi sifat kelas lainnya dan menggambarkan rancangan tersebut dalam bentuk diagram kelas
4. Mahasiswa mampu mengimplementasikan kelas yang mewarisi kelas lainnya.
5. Mahasiswa mampu mempergunakan tipe *visibility* yang tepat untuk mengatur pendalian akses atribut maupun metode dalam suatu kelas*.*
6. **DASAR TEORI**

**B.1 Pewarisan**

Pewarisan merupakan sebuah bentuk “penggunaan kembali” (*reusability*); dimana kelas baru dibuat dari kelas yang pernah ada yang (biasanya) ditambah fasilitasnya. Kelas pokok disebut dengan ***superclass***, sedangkan kelas turunannya disebut dengan ***subclass***. Setiap kelas turunan dapat menjadi kelas pokok (induk) untuk kelas turunan yang akan datang.

Hubungan pewarisan di dalam diagram UML digambarkan dengan cara sebagai berikut:

|  |
| --- |
| **Kelas Atas** |
|  |
|  |

|  |
| --- |
| **Kelas Bawah** |
|  |
|  |

**Gambar 8.1 Diagram kelas untuk Menyatakan Pewarisan**

Kelas turunan mewarisi seluruh atribut dan metode yang terdapat dalam kelas induknya. Artinya bahwa kelas turunan akan otomatis memiliki atribut dan metode seperti kelas induknya. Namun, khusus untuk metode konstruktor, kelas turunan tidak secara otomatis mewarisi konstruktor kelas induknya kecuali jika digunakan perintah **super** yang akan dibahas dalam subbab B.3 modul ini.

Dalam Java, pewarisan dinyatakan dengan menggunakan perintah **extends** yang mengikuti nama kelas dalam deklarasi kelas turunan. Sebagai contoh, terdapat kelas **Atas** seperti berikut ini:

**public class Atas {**

**Atas() {} //constructor**

**public void cetak1() {**

**System.out.println("Ada di class ATAS ..");**

**}}**

Jika kelas **Bawah** merupakan turunan (*subclass)* dari kelas **Atas**, maka dalam Java dituliskan sebagai berikut:

**public class Bawah extends Atas {**

**public void cetak2() {**

**System.out.println("Ada di class BAWAH ..");**

**}**

**public static void main(String[] args) {**

**Bawah bwh = new Bawah();**

**bwh.cetak1();**

**bwh.cetak2();**

**}**

**}**

Dalam metode main di atas, terdapat penggalan program untuk membuat obyek **bwh** bertipe kelas **Bawah** dan memanggil metode **cetak1** dan **cetak2**. Perhatikan bahwa meskipun di dalam deklarasi kelas **Bawah** **tidak tampak** ada metode **cetak1**, namun obyek **bwh** akan mengenali metode **cetak1** karena metode **cetak1** terdapat dalam kelas induknya (kelas **Atas).**

**Tugas1:** Jalankan program tersebut dan amati hasilnya!

**B.2 Kendali akses (lingkup kelas)**

Akses terhadap atribut dan metode dalam suatu kelas dapat dikendalikan dengan tipe *visibility* berikut ini:

1. *Private*:

* Atribut/metode hanya dapat diakses (*visible*) oleh kelasdimana atribut/metodetersebut didefinisikan.

1. *Protected*:

* Atribut/metodedisembunyikan terhadap kelas *client,* tetapi dapat diakses oleh kelasturunannya. Kelas *client* dari kelas A adalah kelas yang menggunakan/membuat obyek dari kelas A.

1. *Public*:

* Atribut/metodedapat diakses oleh semua kelas yang ada (kelas *client* maupun turunan).

Berikut ini adalah contoh penggunaan atribut dengan tipe *visibility private* dalam kelas **Atas\_priv.**

**public class Atas\_priv {**

**private int i = 5;**

**Atas\_priv() {} //constructor**

**public void cetak1() {**

**System.out.println("Ada di class ATAS ..");**

**System.out.println("i = "+i);**

**}}**

**public class Bawah\_priv extends Atas\_priv {**

**public void cetak2() {**

**System.out.println("Ada di class BAWAH ..");**

**System.out.println("i = "+i);**

**}**

**public static void main(String[] args) {**

**Bawah\_priv bwh = new Bawah\_priv();**

**bwh.cetak1();**

**bwh.cetak2();**

**}}**

**Tugas 2:** Jalankan program tersebut dan amati hasilnya!

Berikut ini adalah contoh penggunaan atribut dengan tipe *visibility protected* dalam kelas **Atas\_pro.**

**public class Atas\_pro {**

**protected int i = 5;**

**Atas\_pub() {} //constructor**

**public void cetak1() {**

**System.out.println("Ada di class ATAS ..");**

**System.out.println("i = "+i);**

**}**

**}**

**public class Bawah\_pro extends Atas\_pro {**

**public void cetak2() {**

**System.out.println("Ada di class BAWAH ..");**

**System.out.println("i = "+i);**

**}**

**public static void main(String[] args) {**

**Bawah\_pro bwh = new Bawah\_pro();**

**bwh.cetak1();**

**bwh.cetak2();**

**}**

**}**

**Tugas 3:** Jalankan program tersebut dan amati hasilnya!

Berikut ini adalah contoh penggunaan atribut dengan tipe *visibility public* dalam kelas **Atas\_pub.**

**public class Atas\_pub {**

**public int i = 5;**

**Atas\_pub() {} //constructor**

**public void cetak1() {**

**System.out.println("Ada di class ATAS ..");**

**System.out.println("i = "+i);**

**}**

**}**

**public class Bawah\_pub extends Atas\_pub {**

**public void cetak2() {**

**System.out.println("Ada di class BAWAH ..");**

**System.out.println("i = "+i);**

**}**

**public static void main(String[] args) {**

**Bawah\_pub bwh = new Bawah\_pub();**

**bwh.cetak1();**

**bwh.cetak2();**

**}**

**}**

**Tugas 4:** Jalankan program tersebut dan amati hasilnya!

1. **LATIHAN**

**Kasus I : Pewarisan kelas Pegawai**

* Suatu perusahaan memiliki banyak pegawai. Pegawai mempunyai atribut int npp, String nama, double gajiPokok. Pegawai dibagi atas 3 jenis pegawai yaitu manager, marketing, dan honorer. Penentuan gaji untuk setiap jenis pegawai tersebut berbeda-beda.Manajer memiliki gaji pokok + tunjangan\_jabatan+gaji lembur. Marketing memiliki gaji pokok ditambah bonus. Sedangkan gaji honorer hanya terdiri dari gaji pokok dan gaji lembur.
* Buatlah rancangan diagram kelas yang sesuai dengan deskripsi kasus tersebut. Manfaatkan kelas - kelas yang pernah Anda buat dalam praktikum sebelumnya.
* Implementasikan diagram kelas tersebut dalam bentuk program. Selanjutnya buatlah kelas main bernama MainPegawai yang dapat melakukan:
  + membuat 1 obyek manajer dan memasukkan data-data lengkap manajer tersebut
  + membuat 1 obyek pegawai marketing dan memasukkan data-data lengkap-nya
  + membuat 1 obyek pegawai honorer dan memasukkan data-data lengkap-nya.
  + Menampilkan data ketiga jenis pegawai tersebut beserta gaji total masing-masing.

**Kasus II : Modifikasi Kasus I**

* Berdasarkan solusi kasus I, buatlah program untuk mengelola data pegawai dengan membentuk Array of Pegawai yang kemudian dipakai untuk mengisi data pegawai yang berstatus sebagai manajer, pemasaran (marketing) maupun honorer. Program memberi fasilitas kepada user untuk memasukkan data pegawai lewat keyboard setelah user menjawab berapa pegawai yang akan dimasukkan dan sebelum data pegawai dimasukkan komputer bertanya kepada user tentang status pegawai yang akan dimasukkan yakni apakah manajer, marketing atau honorer. Program kemudian menampilkan data pegawai dengan format berikut:

DATA PEGAWAI PERUSAHAAN YIC (Yes INDONESIAN CAN)

No NPP Nama Status Gaji Pokok Tunjangan Lain\* Gaji Total

------------------------------------------------------------------------------------------------------ 1. 123 Budi Manajer 2000000 500000 2500000

2. 125 Eny Pemasaran 2000000 1000000 3000000

3. ……………………………………………………..

4. ……………………………………………………..

5. ………………………………………………………

Dst ……………………………………………………………

\*) Tunjangan lain tergantung status pegawai

1. **LAPORAN PRAKTIKUM**

Buatlah laporan dari hasil menjalankan dan menganalisa program Anda dalam modul 4. Laporan memuat:

1. Tujuan praktikum (5%)
2. Listing program (hasil *capture screen*) (10%)
3. Output program (hasil *capture screen*) (10%)
4. Algoritma dari main method. (15%)
5. Analisa program yang berisi:
   * Maksud/tujuan umum program (10%)
   * Diagram kelas (10%)
   * Struktur hubungan antara kelas(-kelas) dan main program (10%)
   * Penjelasan atas tiap baris perintah program dalam tiap kelas (10%)
   * Penjelasan keseluruhan jalannya program (10%)
6. Refleksi yang berisi hasil refleksi pribadi saat mengikuti praktikum modul 8 hingga menyusun laporannya. Bagian ini bisa berisi antara lain:
   1. keberhasilan atau kesulitan yang Anda temui saat praktikum maupun menyusun laporan serta perasaan yang muncul sebagai akibatnya
   2. makna (misal nilai-nilai hidup seperti ketekunan, kejujuran, dsb) yang berhasil Anda petik saat praktikum maupun menyusun laporan
   3. dsb (5%)
7. Daftar pustaka yang dipakai untuk membuat laporan (5%)

**Daftar Pustaka**

Wu, C. Thomas. 2006. *An Introduction to Object-Oriented Programming with Java*. Mc Graw-Hill International Edition. 4th edition.

**Excellent means doing ordinary thing extraordinarily**

👍👍 Selamat bekerja. Baik itu belum cukup jika lebih baik itu mungkin… 👍👍