Lembar Kerja Peserta Didik (LKPD)

* Mata Pelajaran : Pemrograman Berorientasi Obyek
* Pokok Bahasan : Java Fundamental
* Subpokok Bahasan : Enheritance
* Tujuan Pembelajaran :
  + Setelah mempraktikkan cara menerapkan enheritance, siswa dapat menjelaskan kegunaan dari konsep enheritance dengan tepat
  + Melalui proyek yang diberikan, siswa dapat membuat dan menerapkan konsep enheritance untuk menyelesaikan permasalahan dengan tepat

# Kegiatan 1

1. Buat project baru ‘InheritanceProject’
2. Buat package ‘mypackage’
3. Di dalam ‘mypackage’, buatlah class ‘Pegawai’ sebagai berikut

public class Pegawai {

public String noPegawai;

public String nama;

public int masaKerja;

public void cetakData(){

System.out.println("No Pegawai : " + noPegawai);

System.out.println("Nama : " + nama);

System.out.println("Masa Kerja : " + masaKerja + " th");

}

}

1. Identifikasi member (atribut dan method) pada class ‘Pegawai’ tersebut!
2. Buatlah class ‘MainClass’ dengan isi sebagai berikut

public class MainClass {

public static void main(String[] args) {

Pegawai p1 = new Pegawai();

p1.noPegawai = "P01";

p1.nama = "Amir";

p1.masaKerja = 30;

p1.cetakData();

}

}

1. Selanjutnya jalankan MainClass tersebut dan amati outputnya!

# Kegiatan 2

1. Buatlah class ‘PegawaiTetap’ di package ‘mypackage’ dengan isi sebagai berikut

public class PegawaiTetap **extends Pegawai** {

public long gajiPokok;

public long tunjangan;

}

1. Identifikasi member (atribut) dari class ‘PegawaiTetap’!
2. Modifikasi ‘MainClass’ nya menjadi

public class MainClass {

public static void main(String[] args) {

Pegawai p1 = new Pegawai();

p1.noPegawai = "P01";

p1.nama = "Amir";

p1.masaKerja = 30;

p1.cetakData();

**PegawaiTetap pt1 = new PegawaiTetap();**

**pt1.noPegawai = "P02";**

**pt1.nama = "Budi";**

**pt1.masaKerja = 25;**

**pt1.gajiPokok = 3000000;**

**pt1.tunjangan = 500000;**

**pt1.cetakData();**

}

}

1. Dengan perubahan tersebut, apakah muncul error?
2. Amati karakteristik obyek pt1 yang merupakan perwujudan dari class ‘PegawaiTetap’ ! Kemudian bandingkan dengan struktur class ‘PegawaiTetap’!
3. Mengapa obyek pt1 dari class ‘PegawaiTetap’ bisa memiliki atribut ‘noPegawai’, ‘nama’, ‘masaKerja’ dan juga method cetakData() yang sama seperti obyek p1 dari class ‘Pegawai’?
4. Menurut Anda, samakah mekanisme di atas jika bentuk class ‘PegawaiTetap’ nya dibuat seperti ini?

public class PegawaiTetap {

public String noPegawai;

public String nama;

public int masaKerja;

**public long gajiPokok;**

**public long tunjangan;**

public void cetakData(){

System.out.println("No Pegawai : " + noPegawai);

System.out.println("Nama : " + nama);

System.out.println("Masa Kerja : " + masaKerja + " th");

}

}

1. Oleh karena itu, apa kegunaan perintah ‘extends’ di dalam perintah

public class PegawaiTetap **extends Pegawai**

1. Selanjutnya coba modifikasi kembali ‘MainClass’ nya menjadi

public class MainClass {

public static void main(String[] args) {

Pegawai p1 = new Pegawai();

p1.noPegawai = "P01";

p1.nama = "Amir";

p1.masaKerja = 30;

**p1.gajiPokok = 2000000;**

**p1.tunjangan = 300000;**

p1.cetakData();

PegawaiTetap pt1 = new PegawaiTetap();

pt1.noPegawai = "P02";

pt1.nama = "Budi";

pt1.masaKerja = 25;

pt1.gajiPokok = 3000000;

pt1.tunjangan = 500000;

pt1.cetakData();

}

}

1. Apakah muncul error dari ‘MainClass’ di atas?
2. Apa yang bisa disimpulkan dari langkah no 7 dan 8 di atas?

Keterangan:

Apa yang dilakukan pada Kegiatan 1 dan 2 di atas, merupakan penerapan dari konsep pewarisan (inheritance). Inheritance merupakan fitur yang ada dalam PBO. Dalam hal ini, class ‘PegawaiTetap’ merupakan turunan dari class ‘Pegawai’, atau dengan kata lain class ‘PegawaiTetap’ dinamakan **subclass** dan class ‘Pegawai’ dinamakan **super class**.

# Kegiatan 3

1. Buatlah class ‘PegawaiTetapDN’ (pegawai tetap dalam negeri) dengan bentuk sbb:

public class PegawaiTetapDN **extends PegawaiTetap** {

public String noKTP;

}

Dalam hal ini, ‘PegawaiTetapDN’ merupakan subclass dari ‘PegawaiTetap’.

1. Selanjutnya di ‘MainClass’ buatlah sebuah obyek dari class PegawaiTetapDN, misalkan ‘ptdn1’!
2. Lakukan identifikasi, apakah ‘ptdn1’ ini masih memiliki atribut ‘noPegawai’, ‘nama’, ‘masaKerja’, dan juga method cetakData()?
3. Lakukan identifikasi juga, apakah ‘ptdn1’ ini memiliki atribut ‘gajiPokok’ dan ‘tunjangan’?
4. Apa yang bisa disimpulkan dari kegiatan ini?

# Kegiatan 4

1. Modifikasilah class ‘PegawaiTetapDN’ menjadi

public **final** class PegawaiTetapDN extends PegawaiTetap {

public String noKTP;

}

1. Selanjutnya coba buat class dengan nama ‘PegawaiX’ yang merupakan turunan dari ‘PegawaiTetapDN’.
2. Perhatikan class ‘PegawaiX’ tersebut, apakah terjadi error?
3. Apa kesimpulan kalian terkait dengan kegunaan perintah ‘final’ di atas?

# Latihan

1. Buatlah class ‘PegawaiTidakTetap’ yang merupakan subclass (turunan) dari class ‘Pegawai’. Terdapat atribut khusus untuk class ini yaitu hanya ‘gajiPokok’ dengan tipe data long.
2. Buatlah class ‘PegawaiTetapLN’ (pegawai tetap dari luar negeri) yang merupakan subclass dari class ‘PegawaiTetap’. Terdapat atribut khusus untuk class ini yaitu ‘noPaspor’ dengan tipe data string.

# Kesimpulan

Berdasarkan kegiatan-kegiatan di atas, coba simpulkan beberapa hal berikut ini:

1. Apa itu inheritance?
2. Apa manfaat inheritance dalam PBO?
3. Apa itu subclass?
4. Apa itu superclass?
5. Bagaimana sifat subclass jika dibandingkan dengan superclass?
6. Apa kegunaan dari keyword ‘final’ dalam pendefinisian class