

MODUL 6

INHERITANCE (PEWARISAN)

1. KOMPETENSI

1. Memahami konsep dasar inheritance atau pewarisan.
2. Mampu memahami jenis-jenis inheritance.
3. Mampu mengimplementasikan konsep inheritance pada Bahasa pemrograman Java

2. PENDAHULUAN

Inheritance atau **pewarisan sifat** merupakan suatu cara untuk menurunkan suatu class yang lebih umum menjadi suatu class yang lebih spesifik. Inheritance adalah salah satu ciri utama suatu bahasa program yang berorientasi pada objek. Inti dari pewarisan adalah sifat reusable dari konsep object oriented. Setiap **subclass** akan “mewarisi” sifat dari **superclass** selama bersifat protected ataupun public.

Dalam inheritance terdapat dua istilah yang sering digunakan. Kelas yang menurunkan disebut kelas dasar (**base class/super class**), sedangkan kelas yang diturunkan disebut kelas turunan (**derived class/sub class/child class**). Di dalam Java untuk mendeklarasikan suatu class sebagai subclass dilakukan dengan cara menambahkan kata kunci **extends** setelah deklarasi nama class, kemudian diikuti dengan nama parent class-nya. Kata kunci extends tersebut memberitahu kompiler Java bahwa kita ingin melakukan perluasan class. Berikut adalah contoh deklarasi inheritance.

```
public class B extends A {  
    ...  
}
```

Contoh diatas memberitahukan kompiler Java bahwa kita ingin meng-extend class A ke class B. Dengan kata lain, class B adalah subclass (class turunan) dari class A, sedangkan class A adalah parent class dari class B.

Karakteristik pada super class akan dimiliki juga oleh subclassnya. Terdapat 5 bentuk pewarisan: single inheritance, multilevel inheritance, hierarchical inheritance, hybrid inheritance, dan multiple inheritance.

1. Single Inheritance

Single inheritance adalah Suatu class yang hanya mempunyai satu parent class. Contoh:



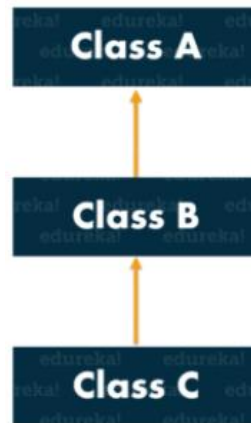
Gambar 1. Contoh Single Inheritance

Berdasarkan Gambar 1 dapat diketahui bahwa class B merupakan subclass yang mempunyai satu parent yaitu class A sehingga disebut single inheritance.

2. Multilevel Inheritance

Multilevel inheritance adalah Suatu subclass bisa menjadi superclass bagi class yang lain.

Contoh:

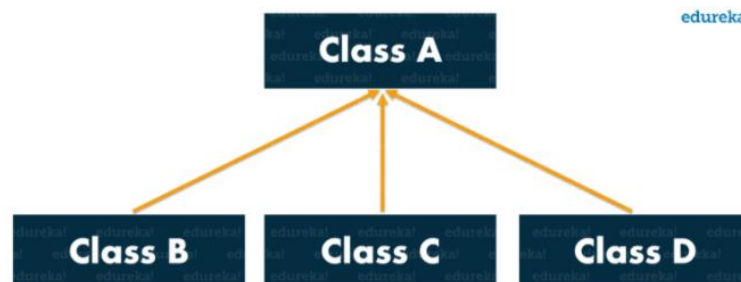


Gambar 2. Contoh Multilevel Inheritance

Berdasarkan Gambar 2 diatas dapat dilihat bahwa class B merupakan subclass dari class A, sehingga dalam hal ini class A adalah superclass dan class B adalah subclass. Kemudian class B yang awalnya merupakan subclass mempunyai subclass lagi yaitu class C sehingga class B menjadi superclass dari class C, begitu juga seterusnya jika class C memilki subclass lagi.

3. Hierarchical Inheritance

Hierarchical inheritance merupakan contoh pewarisan yang terjadi ketika sebuah kelas memiliki lebih dari satu kelas turunan (*subclass*) atau dengan kata lain, lebih dari satu kelas turunan memiliki kelas induk yang sama. Konsep ini sesuai dengan contoh di dunia nyata, dimana satu orangtua bisa memiliki lebih dari satu anak. Contoh bentul hierarchical inheritance dapat dilihat pada Gambar 3.

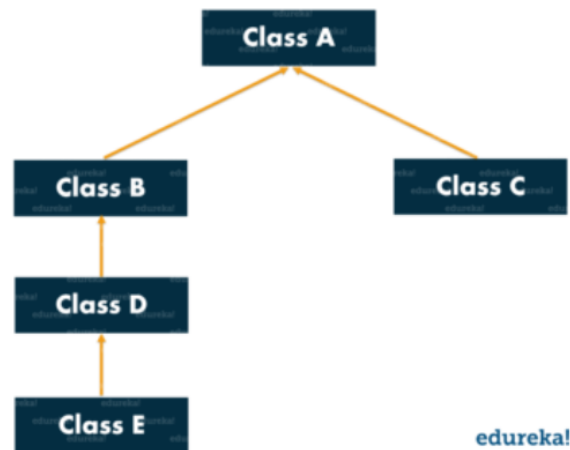


Gambar 3. Contoh Hierarchical Inheritance

Pada Gambar 3 dapat dilihat bahwa class A memiliki 3 subclass, yaitu class B, class C, dan class D. Dengan demikian, class B, class C, dan class D memiliki superclass yang sama, dan ada kemungkinan ketiga class subclass tersebut memiliki atribut dan behaviour yang sama dari class A.

4. Hybrid Inheritance

Hybrid inheritance merupakan jenis pewarisan yang merupakan kombinasi dari dua atau lebih jenis pewarisan. Contoh hybrid inheritance dapat dilihat pada Gambar 4.

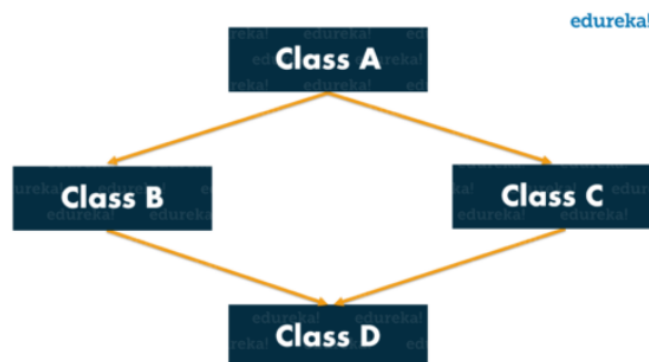


Gambar 4. Contoh Hybrid Inheritance

Pada Gambar 4 dapat dilihat bahwa bentuk pewarisan tersebut merupakan kombinasi dari hierarchical dan multilevel inheritance. Hierarchical inheritance ditunjukkan oleh Class A, class B, dan class C. Sedangkan multilevel inheritance ditunjukkan oleh class B, class D, dan class E.

Terdapat beberapa aturan pada pewarisan menggunakan Bahasa pemrograman Java, antara lain:

1. Multiple inheritance tidak dapat digunakan pada Bahasa pemrograman Java. Multiple inheritance mengacu pada proses dimana satu kelas turunan mencoba untuk memperluas (extend) lebih dari satu kelas induk. Contoh multiple inheritance dapat dilihat pada Gambar 5.

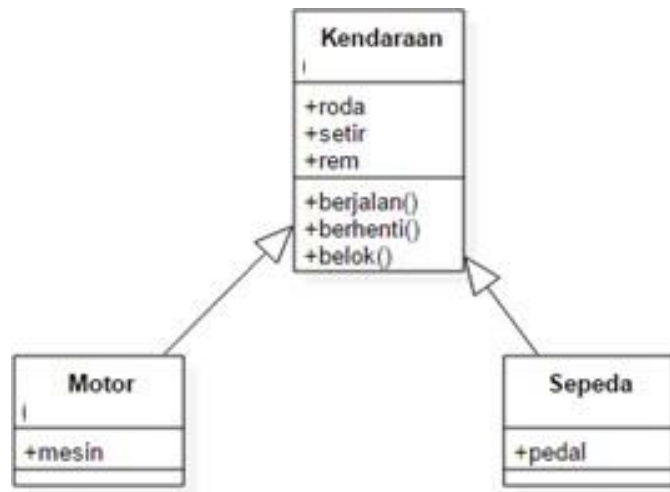


Gambar 5. Contoh multiple inheritance

Misal terdapat method `show()` pada kelas B dan C dengan fungsi yang berbeda. Kemudian kelas D meng-extend kelas B dan C. Ketika objek dari kelas D mencoba memanggil method `show()`, kompilator akan bingung method di kelas mana yang akan dieksekusi (dari kelas B atau C). Sehingga hal ini mengarah pada ambiguitas.

2. Cyclic inheritance tidak dapat digunakan di Java. Cyclic inheritance merupakan jenis inheritance dimana sebuah kelas memperluas (extend) dirinya sendiri. Jenis ini tidak diizinkan oleh Java karena tidak ada peluang untuk memperluas kelas Object.
3. Atribut dan method dengan access modifier `Private` tidak diwariskan.
4. Constructor tidak dapat diwariskan. Jika ingin menggunakan constructor kelas induk, harus menggunakan perintah `super()` pada constructor kelas anak.

Pada class diagram, pewarisan digambarkan dengan sebuah garis tegas, dengan segitiga di ujungnya. Class yang dekat pada segitiga merupakan superclass, sedangkan class yang jauh dari segitiga merupakan subclass. Untuk membentuk sebuah subclass, keyword “extends” digunakan (lihat contoh pada sesi “Implementasi Pewarisan”). Berikut ini adalah contoh class diagram dari pewarisan:

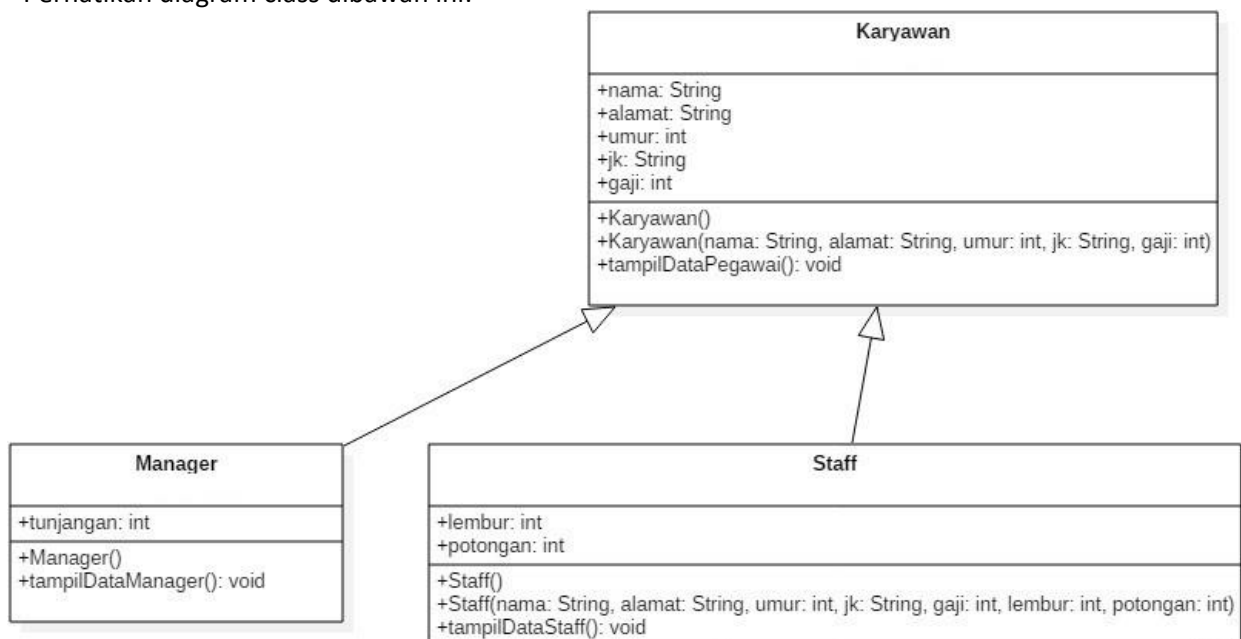


Gambar 6 Contoh class diagram dalam inheritance

3. PERCOBAAN 1

A. TAHAPAN PERCOBAAN

Perhatikan diagram class dibawah ini:





1. Buatlah class Karyawan

```
11  /*
12  *
13  */
14  public class Karyawan {
15      public String nama, alamat, jk;
16      public int umur, gaji;
17
18      public Karyawan() {
19      }
20
21      public Karyawan(String nama, String alamat, String jk, int umur, int gaji) {
22          this.nama=nama;
23          this.alamat=alamat;
24          this.jk=jk;
25          this.umur=umur;
26          this.gaji=gaji;
27      }
28
29      public void tampilDataKaryawan() {
30          System.out.println("Nama          =" +nama);
31          System.out.println("Alamat       =" +alamat);
32          System.out.println("Jenis Kelamin =" +jk);
33          System.out.println("Umur         =" +umur);
34          System.out.println("Gaji         =" +gaji);
35      }
36  }
```

2. Buatlah class Manager

```
10  * @author WINDOWS 10
11  */
12  public class Manager extends Karyawan{
13      public int tunjangan;
14
15      public Manager() {
16      }
17
18      public void tampilDataManager() {
19          super.tampilDataKaryawan();
20          System.out.println("Tunjangan      =" +tunjangan);
21          System.out.println("Total Gaji     =" + (super.gaji+tunjangan));
22      }
23
24  }
```



3. Buatlah class Staff

```

10  |  * @author WINDOWS 10
11  |  */
12  |  public class Staff extends Karyawan {
13  |      public int lembur, potongan;
14  |
15  |      public Staff(){
16  |      }
17  |
18  |      public Staff(String nama, String alamat, String jk, int umur, int gaji, int lembur, int potongan){
19  |          super(nama, alamat, jk, umur, gaji);
20  |          this.lembur=lembur;
21  |          this.potongan=potongan;
22  |      }
23  |
24  |      public void tampilDataStaff(){
25  |          super.tampilDataKaryawan();
26  |          System.out.println("Lembur          =" +lembur);
27  |          System.out.println("Potongan       =" +potongan);
28  |          System.out.println("Total Gaji      =" +(gaji+lembur-potongan));
29  |      }
30  |  }

```

4. Buatlah class inheritance1

```

11  |  */
12  |  public class Inheritance1 {
13  |
14  |      /**
15  |       * @param args the command line arguments
16  |       */
17  |      public static void main(String[] args) {
18  |          // TODO code application logic here
19  |          Manager M = new Manager();
20  |          M.nama="Vivin";
21  |          M.alamat="Jl. Vinolia";
22  |          M.umur=25;
23  |          M.jk="Perempuan";
24  |          M.gaji=3000000;
25  |          M.tunjangan=1000000;
26  |          M.tampilDataManager();
27  |
28  |          Staff S = new Staff();
29  |          S.nama="Lestari";
30  |          S.alamat="Malang";
31  |          S.umur=25;
32  |          S.jk="Perempuan";
33  |          S.gaji=2000000;
34  |          S.lembur=500000;
35  |          S.potongan=250000;
36  |          S.tampilDataStaff();
37  |      }
38  |
39  |  }
40  |

```

5. Jalankan program, maka tampilanya adalah sebagai berikut:

```

Output - inheritance1 (run)

run:
Nama      =Vivin
Alamat    =Jl. Vinolia
Jenis Kelamin =Perempuan
Umur      =25
Gaji      =3000000
Tunjangan =1000000
Total Gaji =4000000
Nama      =Lestari
Alamat    =Malang
Jenis Kelamin =Perempuan
Umur      =25
Gaji      =2000000
Lembur    =500000
Potongan  =250000
Total Gaji =2250000
BUILD SUCCESSFUL (total time: 0 seconds)

```

B. PERTANYAAN

1. Sebutkan class mana yang termasuk super class dan sub class dari percobaan 1 diatas!

.....

.....

.....

.....

2. Kata kunci apakah yang digunakan untuk menurunkan suatu class ke class yang lain?

.....

.....

.....

.....

3. Perhatikan kode program pada class Manager, atribut apa saja yang dimiliki oleh class tersebut? Sebutkan atribut mana saja yang diwarisi dari class Karyawan!

.....

.....

.....

.....

4. Jelaskan kata kunci super pada potongan program dibawah ini yang terdapat pada class Manager!

```
System.out.println("Total Gaji      =" + (super.gaji+tunjangan));
```

.....

.....

.....

.....

5. Program pada percobaan 1 diatas termasuk dalam jenis inheritance apa? Jelaskan alasannya!

.....

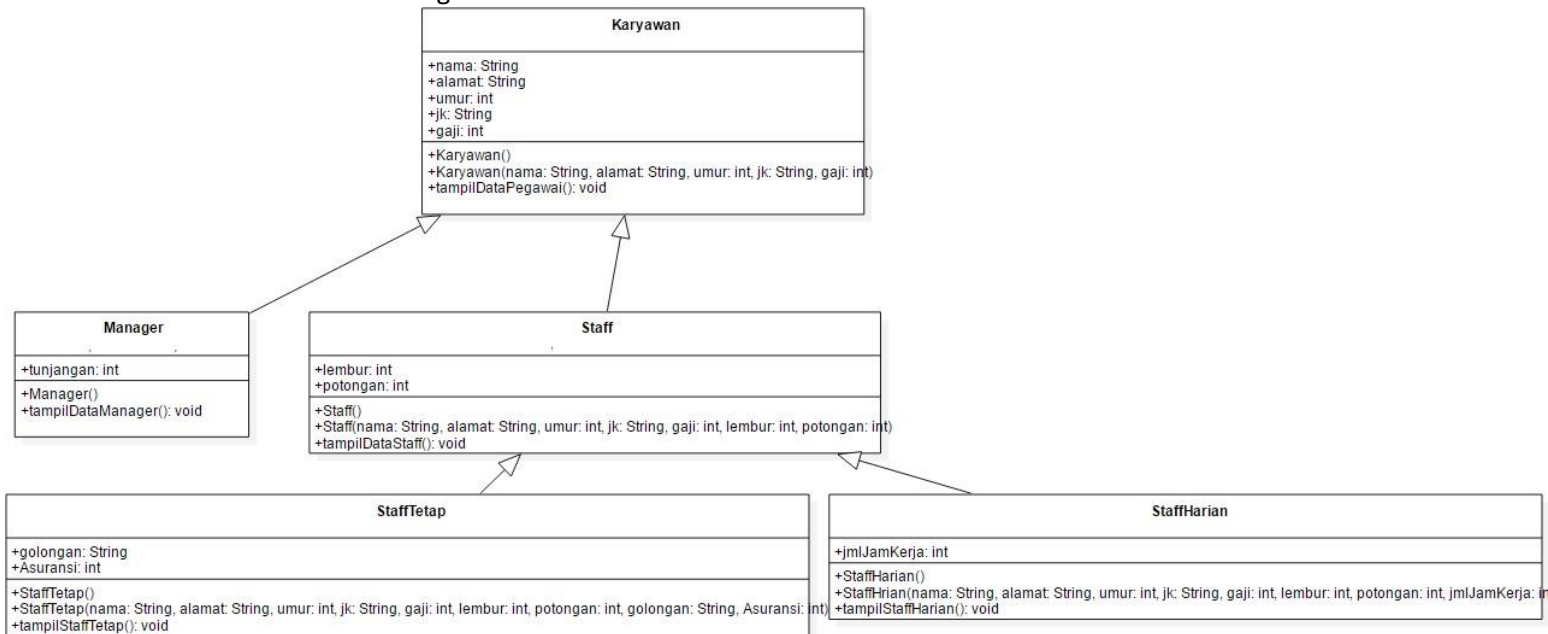
.....

.....

4. PERCOBAAN 2

1. TAHAPAN PERCOBAAN

Perhatikan digram class dibawah ini



- Berdasarkan program yang sudah anda buat pada percobaan 1 sebelumnya tambahkan dua class yaitu class StaffTetap dan class StaffHarian. Kode Programnya adalah sebagai berikut

Class StaffTetap.java

```

10  // @author WINDOWS 10
11  /*
12  public class StaffTetap extends Staff{
13      public String golongan;
14      public int asuransi;
15
16      public StaffTetap() {
17
18      }
19      public StaffTetap(String nama, String alamat, String jk, int umur,
20          int gaji, int lembur, int potongan, String golongan, int asuransi){
21          super(nama, alamat, jk, umur, gaji, potongan, lembur);
22          this.golongan=golongan;
23          this.asuransi=asuransi;
24      }
25      public void tampilStaffTetap() {
26          System.out.println("=====Data Staff Tetap=====");
27          super.tampilDataStaff();
28          System.out.println("Golongan          =" + golongan);
29          System.out.println("Jumlah Asuransi =" + asuransi);
30          System.out.println("Gaji Bersih      =" + (gaji+lembur-potongan-asuransi));
31      }
32
33  }
  
```




Class StaffHarian.java

```

11  L  */
12  public class StaffHarian extends Staff{
13      public int jmlJamKerja;
14
15  [-] public StaffHarian(){
16
17  }
18  public StaffHarian(String nama, String alamat, String jk, int umur,
19      int gaji, int lembur, int potongan,int jmlJamKerja){
20      super(nama,alamat,jk,umur,gaji,lembur,potongan);
21      this.jmlJamKerja=jmlJamKerja;
22  }
23  [-] public void tampilStaffHarian(){
24      System.out.println("=====Data Staff Harian=====");
25      super.tampilDataStaff();
26      System.out.println("Jumlah Jam Kerja =" +jmlJamKerja);
27      System.out.println("Gaji Bersih      =" +(gaji*jmlJamKerja+lembur-potongan));
28  }
29
30  }
31

```

- Setelah membuat dua class diatas kemudian edit class inheritance1.java menjadi sebagai berikut:

```

11  L  */
12  public class Inheritance1 {
13
14  [-] /**
15      * @param args the command line arguments
16      */
17  [-] public static void main(String[] args) {
18      // TODO code application logic here
19      StaffTetap ST= new StaffTetap("Budi", "Malang", "Lakilaki", 20, 2000000, 250000, 200000, "2A", 100000);
20      ST.tampilStaffTetap();
21
22
23      StaffHarian SH = new StaffHarian("Indah", "Malang", "Perempuan", 27, 10000, 100000, 50000, 100);
24      SH.tampilStaffHarian();
25  }
26
27  }
28

```

- Jalankan program maka tampilannya adalah sebagai berikut:

```

Output - inheritance1 (run)

run:
=====Data Staff Tetap=====
Nama           =Budi
Alamat         =Malang
Jenis Kelamin   =Lakilaki
Umur           =20
Gaji           =2000000
Lembur         =200000
Potongan       =250000
Total Gaji     =1950000
Golongan       =2A
Jumlah Asuransi=100000
Gaji Bersih    =1850000
=====Data Staff Harian=====
Nama           =Indah
Alamat         =Malang
Jenis Kelamin   =Perempuan
Umur           =27
Gaji           =10000
Lembur         =100000
Potongan       =50000
Total Gaji     =60000
Jumlah Jam Kerja=100
Gaji Bersih    =1050000
BUILD SUCCESSFUL (total time: 0 seconds)

```

2. PERTANYAAN

- Berdasarkan class diatas manakah yang termasuk single inheritance dan mana yang termasuk multilevel inheritance?

.....

.....

.....

- Perhatikan kode program class StaffTetap dan StaffHarian, atribut apa saja yang dimiliki oleh class tersebut? Sebutkan atribut mana saja yang diwarisi dari class Staff!

.....

.....

.....

- Apakah fungsi potongan program berikut pada class StaffHarian

```
super(nama, alamat, jk, umur, gaji, lembur, potongan);
```

.....

.....

.....

- Apakah fungsi potongan program berikut pada class StaffHarian

```
super.tampilDataStaff();
```

9. Perhatikan kode program dibawah ini yang terdapat pada class StaffTetap

```
System.out.println("Gaji Bersih      "+(gaji+lembur-potongan-asuransi));
```

Terlihat dipotongan program diatas atribut gaji, lembur dan potongan dapat diakses langsung. Kenapa hal ini bisa terjadi dan bagaimana class StaffTetap memiliki atribut gaji, lembur, dan potongan padahal dalam class tersebut tidak dideklarasikan atribut gaji, lembur, dan potongan?

5. TUGAS

1. Buatlah sebuah program dengan konsep pewarisan seperti pada class diagram berikut ini. Kemudian buatlah instansiasi objek untuk menampilkan data pada class Mac, Windows dan PC!.

