

JOBSHEET 7

INHERITANCE & POLYMORPHISM

Nama : Rizqi Rohmatul Huda
Kelas : 2G – TI
No.Absen : 26
NIM : 2141720264

3. PERCOBAAN 1 (extends)

A. TAHAPAN PERCOBAAN

1. Buatlah sebuah parent class dengan nama Pegawai. Lalu buat constructor tanpa parameter dengan baris kode sebagai berikut:

```
public class Pegawai {  
  
    public Pegawai(){  
        System.out.println("Objek dari class Pegawai dibuat");  
    }  
}
```

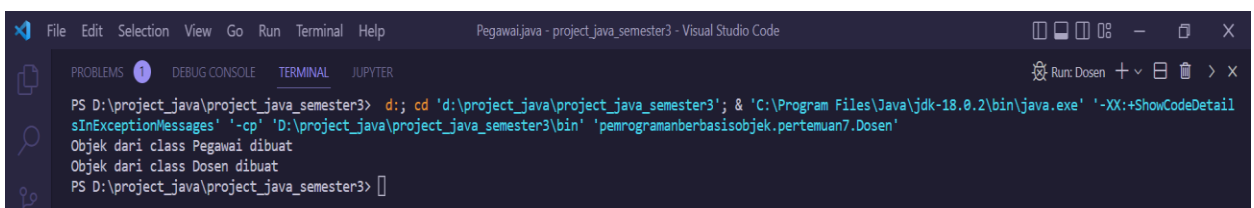
2. Buatlah subclass dari class Pegawai dengan nama Dosen, kemudian buat juga constructor tanpa parameter dengan baris kode berikut:

```
class Dosen extends Pegawai{  
  
    public Dosen(){  
        System.out.println("Objek dari class Dosen dibuat");  
    }  
}
```

3. Buatlah main class, misal InheritanceDemo.java, lakukan instansiasi objek baru bernama dosen1 dari class Dosen sebagai berikut:

```
public static void main(String[] args) {  
    Dosen dosen1 = new Dosen();  
}
```

4. Run programnya kemudian amati hasilnya.



```
File Edit Selection View Go Run Terminal Help  
Pegawai.java - project_java_semester3 - Visual Studio Code  
PROBLEMS DEBUG CONSOLE TERMINAL JUPYTER  
Run: Dosen + - [ ] > X  
PS D:\project_java\project_java_semester3> d:; cd 'd:\project_java\project_java_semester3'; & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\project_java\project_java_semester3\bin' 'pemrogramanberbasisobjek.pertemuan7.Dosen'  
Objek dari class Pegawai dibuat  
Objek dari class Dosen dibuat  
PS D:\project_java\project_java_semester3> [ ]
```

B. PERTANYAAN

1. Pada percobaan 1 diatas, tentukan child class dan parent class!

Jawab :

- a Child class : Dosen
- b Parent class : Pegawai

2. Kata kunci apa yang membuat child class dan parent class tersebut memiliki relasi?

Jawab : Menggunakan kata kunci **extends**

3. Berdasarkan hasil yang ditampilkan oleh program, ada berapa constructor yang dieksekusi? Constructor class mana yang lebih dulu dieksekusi?

Jawab : ada 2 constructor yang dieksekusi, constructor class Pegawai yang dieksekusi terlebih dahulu.

4. PERCOBAAN 2 (Pewarisan)

A. TAHAPAN PERCOBAAN

1. Tambahkan atribut nip, nama, dan gaji serta method getInfo() pada class Pegawai

```
public class Pegawai {  
    public String nip;  
    public String nama;  
    public double gaji;  
  
    public Pegawai(){  
        System.out.println("Objek dari class Pegawai dibuat");  
    }  
  
    public String getInfo(){  
        String info = "";  
        info += "NIP\t\t\t\t\t: " + nip + "\n";  
        info += "Nama\t\t\t\t\t: " + nama + "\n";  
        info += "Gaji\t\t\t\t\t: " + gaji + "\n";  
  
        return info;  
    }  
}
```

2. Tambahkan pula atribut NIDN pada class Dosen

```
class Dosen extends Pegawai{  
    public String nidn;  
  
    public Dosen(){  
        System.out.println("Objek dari class Dosen dibuat");  
    }  
}
```

3. Pada class InheritanceDemo.java tuliskan baris kode berikut:

```
public static void main(String[] args) {  
    Dosen dosen1 = new Dosen();  
}
```

```

        dosen1.nama = "Yansy Ayuningtyas";
        dosen1.nip = "34329837";
        dosen1.gaji = 3000000;
        dosen1.nidn = "1989432439";

        System.out.println(dosen1.getInfo());
    }
}

```

4. Run program kemudian amati hasilnya

```

PS D:\project_java\project_java_semester3> & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\project_java\project_java_semester3\bin' 'pemrogramanberbasisobjek.pertemuan7.Dosen'
Objek dari class Pegawai dibuat
Objek dari class Dosen dibuat
NIP          : 34329837
Nama         : Yansy Ayuningtyas
Gaji         : 3000000.0
PS D:\project_java\project_java_semester3>

```

B. PERTANYAAN

1. Pada percobaan 2 diatas, apakah program dapat berhasil dijalankan ataukah terjadi error?
Jawab : Iya, program berhasil dijalankan

2. Jika program berhasil dijalankan, mengapa tidak terjadi error pada assignment/pengisian nilai atribut nip, gaji, dan NIDN pada object dosen1 padahal tidak ada deklarasi ketiga atribut tersebut pada class Dosen?

Jawab : karena terdapat keyword **extends** sebagai deklarasi. keyword extends tersebut memberitahu kompiler Java bahwa kita ingin melakukan **extension/ perluasan** class.

3. Jika program berhasil dijalankan, mengapa tidak terjadi error pada pemanggilan method getInfo() oleh object dosen1 padahal tidak ada deklarasi method getInfo() pada class Dosen?

Jawab : Karena jika class Dosen meng-extends dari class Pegawai maka akan otomatis atribut dan methodnya akan dideklarasikan (otomatis mewarisi atribut dan method dari parent class).

5. PERCOBAAN 3 (Hak akses)

A. TAHAPAN PERCOBAAN

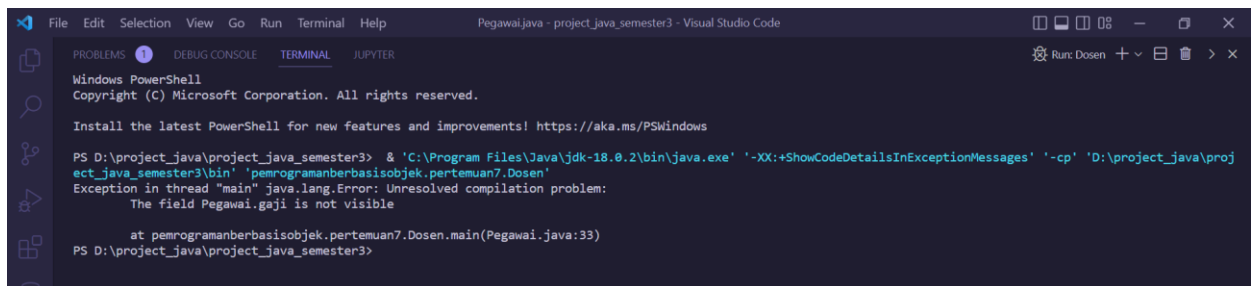
1. Modifikasi access level modifier pada atribut gaji menjadi private pada class Pegawai.java

```

public class Pegawai {
    public String nip;
    public String nama;
    private double gaji;
}

```

2. Run program kemudian amati hasilnya.



```
File Edit Selection View Go Run Terminal Help
Pegawai.java - project_java_semester3 - Visual Studio Code

PROBLEMS 1 DEBUG CONSOLE TERMINAL JUPYTER
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\project_java\project_java_semester3> & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\project_java\proj
ect_java_semester3\bin' 'pemrogramanberbasisobjek.pertemuan7.Dosen'
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    The field Pegawai.gaji is not visible
    at pemrogramanberbasisobjek.pertemuan7.Dosen.main(Pegawai.java:33)
PS D:\project_java\project_java_semester3>
```

3. Ubah access level modifier atribut gaji menjadi protected kemudian pindah class Pegawai ke package baru, misalnya "testpackage".

```
package pemrogramanberbasisobjek.pertemuan7.testpackage;

public class Pegawai {
    public String nip;
    public String nama;
    protected double gaji;
```

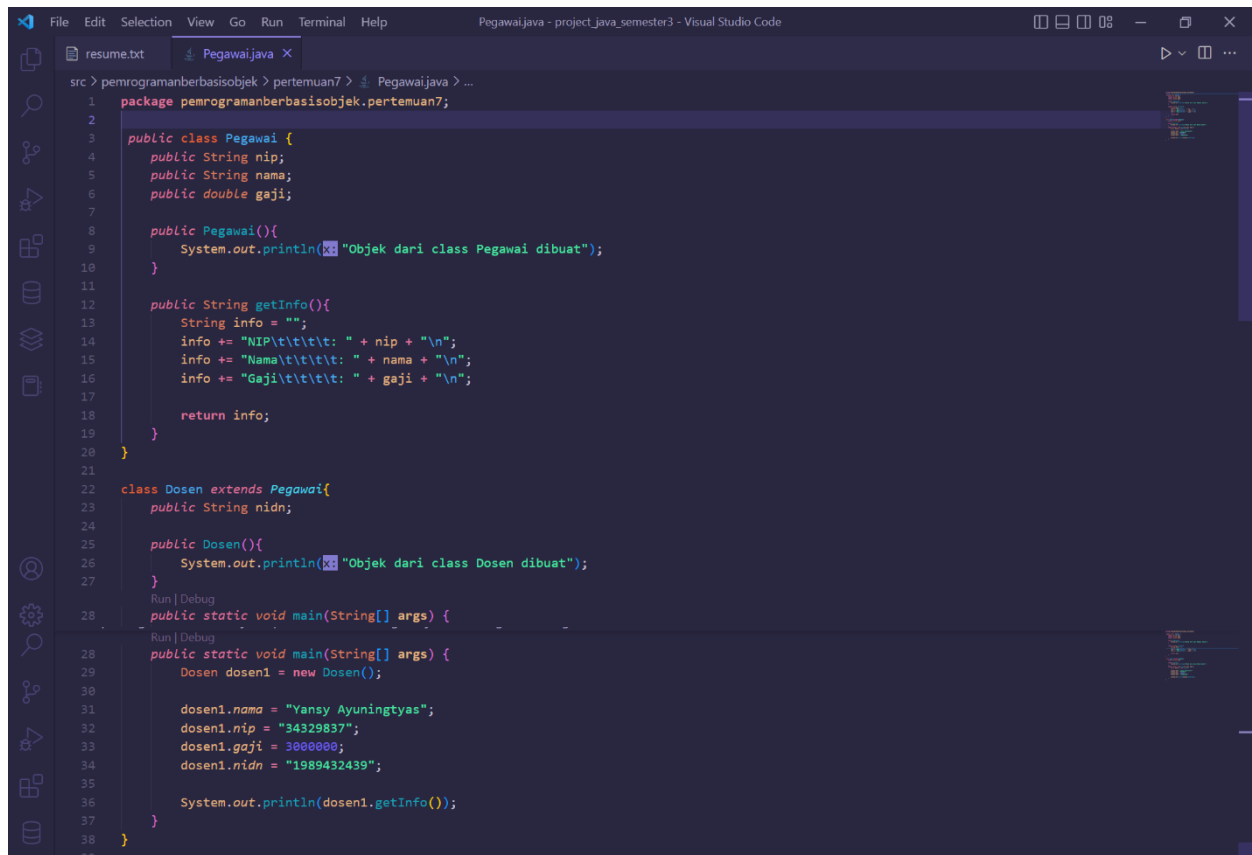
4. Import class Pegawai dari testpackage pada class Dosen.

```
package pemrogramanberbasisobjek.pertemuan7.inheritance;
import pemrogramanberbasisobjek.pertemuan7.testpackage.Pegawai;
```

5. Akses atribut gaji pada class Dosen dengan coba mencetak atribut gaji pada constructor Dosen

```
public Dosen(){
    System.out.println(gaji);
    System.out.println("Objek dari class Dosen dibuat");
}
```

6. Ubah kembali access level modifier menjadi public dan kembalikan class Pegawai ke package semula.



```
1 package pemrogramanberbasisobjek.pertemuan7;
2
3 public class Pegawai {
4     public String nip;
5     public String nama;
6     public double gaji;
7
8     public Pegawai(){
9         System.out.println("Objek dari class Pegawai dibuat");
10    }
11
12    public String getInfo(){
13        String info = "";
14        info += "NIP\t\t\t\t: " + nip + "\n";
15        info += "Nama\t\t\t\t: " + nama + "\n";
16        info += "Gaji\t\t\t\t: " + gaji + "\n";
17
18        return info;
19    }
20 }
21
22 class Dosen extends Pegawai{
23     public String nidn;
24
25     public Dosen(){
26         System.out.println("Objek dari class Dosen dibuat");
27     }
28
29     public static void main(String[] args) {
30         Dosen dosen1 = new Dosen();
31
32         dosen1.nama = "Yansy Ayuningtyas";
33         dosen1.nip = "34329837";
34         dosen1.gaji = 3000000;
35         dosen1.nidn = "1989432439";
36
37         System.out.println(dosen1.getInfo());
38     }
39 }
```

B. PERTANYAAN

1. Pada langkah 1 di atas, terjadi error karena object dosen1 tidak dapat mengakses atribut gaji. Padahal gaji merupakan atribut Pegawai yang merupakan parent class dari Dosen. Mengapa hal ini dapat terjadi?

Jawab : Karena pada atribut gaji akses modifiernya diset sebagai private, sehingga pada saat diprivate hanya dapat mengakses pada class yang sama, meskipun gaji merupakan atribut Pegawai yang merupakan parent class dari Dosen.

2. Pada langkah 5, setelah class Pegawai berpindah ke package yang berbeda, class Dosen masih dapat mengakses atribut gaji. Mengapa?

Jawab : Karena pada atribut gaji akses modifiernya diubah menjadi protected, jadi masih dapat mengakses class Pegawai setelah berpindah ke package yang berbeda

3. Berdasarkan percobaan tersebut, bagaimana menentukan atribut dan method yang akan diwariskan oleh parent class ke child class?

Jawab : Dengan cara pada semua atribut dan method akses modifiernya tidak diset private. Jadi akses modifier boleh public, protected boleh diluar package asal merupakan subclass (pewarisan terhadap class nya), dan no modifier asal di dalam package yang sama.

6. PERCOBAAN 4 (Super - atribut)

A. TAHAPAN PERCOBAAN

1. Butlah method getAllInfo() pada class Dosen

```
public String getAllInfo(){
    String info = "";
    info += "NIP\t\t\t\t: " + nip + "\n";
    info += "Nama\t\t\t\t: " + nama + "\n";
    info += "Gaji\t\t\t\t: " + gaji + "\n";
    info += "NIDN\t\t\t\t: " + nidn + "\n";

    return info;
}
```

2. Lakukan pemanggilan method getAllInfo() oleh object dosen1 pada class InheritanceDemo.java

```
public static void main(String[] args) {
    Dosen dosen1 = new Dosen();

    dosen1.nama = "Yansy Ayuningtyas";
    dosen1.nip = "34329837";
    dosen1.gaji = 3000000;
    dosen1.nidn = "1989432439";

    System.out.println(dosen1.getAllInfo());
}
```

3. Run program kemudian amati hasilnya

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\project_java\project_java_semester3> & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\project_java\project_java_semester3\bin' 'pemrogramanberbasisobjek.pertemuan7.InheritanceDemo'
Objek dari class Pegawai dibuat
Objek dari class Dosen dibuat
NIP      : 34329837
Nama     : Yansy Ayuningtyas
Gaji     : 3000000.0
NIDN     : 1989432439

PS D:\project_java\project_java_semester3>
```

4. Lakukan modifikasi method getAllInfo() pada class Dosen

```
public String getAllInfo(){
    String info = "";
    info += "NIP\t\t\t\t: " + this.nip + "\n";
    info += "Nama\t\t\t\t: " + this.nama + "\n";
    info += "Gaji\t\t\t\t: " + this.gaji + "\n";
    info += "NIDN\t\t\t\t: " + this.nidn + "\n";

    return info;
}
```

5. Run program kemudian bandingkan hasilnya dengan langkah no 2.

```

PS D:\project_java\project_java_semester3> & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\project_java\project_java_semester3\bin' 'pemrogramanberbasisobjek.pertemuan7.InheritanceDemo'
Objek dari class Pegawai dibuat
Objek dari class Dosen dibuat
NIP          : 34329837
Nama         : Yansy Ayuningtyas
Gaji         : 3000000.0
NIDN        : 1989432439

```

6. Lakukan modifikasi method `getAllInfo()` pada class `Dosen` kembali

```

public String getAllInfo(){
    String info = "";
    info += "NIP\t\t\t\t: " + super.nip + "\n";
    info += "Nama\t\t\t\t: " + super.nama + "\n";
    info += "Gaji\t\t\t\t: " + super.gaji + "\n";
    info += "NIDN\t\t\t\t: " + super.nidn + "\n";

    return info;
}

```

7. Run program kemudian bandingkan hasilnya dengan program pada no 1 dan no 4.

```

PS D:\project_java\project_java_semester3> & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\project_java\project_java_semester3\bin' 'pemrogramanberbasisobjek.pertemuan7.Dosen'
Objek dari class Pegawai dibuat
Objek dari class Dosen dibuat
NIP          : 34329837
Nama         : Yansy Ayuningtyas
Gaji         : 3000000.0

PS D:\project_java\project_java_semester3>

```

8. Lakukan modifikasi method `getAllInfo()` pada class `Dosen` kembali

```

public String getAllInfo(){
    String info = "";
    info += "NIP\t\t\t\t: " + super.nip + "\n";
    info += "Nama\t\t\t\t: " + super.nama + "\n";
    info += "Gaji\t\t\t\t: " + super.gaji + "\n";
    info += "NIDN\t\t\t\t: " + this.nidn + "\n";

    return info;
}

```

9. Run program kemudian bandingkan hasilnya dengan program pada no 2 dan no 4.

```

PS D:\project_java\project_java_semester3> & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\project_java\project_java_semester3\bin' 'pemrogramanberbasisobjek.pertemuan7.Dosen'
Objek dari class Pegawai dibuat
Objek dari class Dosen dibuat
NIP          : 34329837
Nama         : Yansy Ayuningtyas
Gaji         : 3000000.0

```

```
P> d:; cd 'd:\project_java\project_java_semester3'; & 'C:\Program Files\Java\jdk-18
.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\project_java\
project_java_semester3\bin' 'pemrogramanberbasisobjek.pertemuan7.Dosen'
Objek dari class Pegawai dibuat
Objek dari class Dosen dibuat
NIP                : 34329837
Nama               : Yansy Ayuningtyas
Gaji               : 3000000.0

PS D:\project_java\project_java_semester3>
```

B. PERTANYAAN

1. Apakah terdapat perbedaan hasil nama, nip, dan gaji yang ditampilkan pada program 1,4,dan 8? Mengapa?

Jawab : iya, ada

- Pada program 1 dan 4 nip, nama, gaji, nidn dapat ditampilkan dengan lengkap
- Pada program 8 nidn tidak ditampilkan

Terjadi perbedaan pada program 8 karena pada saat atribut memakai keyword this pada nidn, maka nidn tidak ditampilkan, keyword this akan merujuk class child. Sedangkan pada class parent tidak ada atribut nidn, jadi nidn tidak akan ditampilkan.

Lalu pada program 1 dan program 4 akan memiliki kesamaan karena pada program 1 ketika tidak memakai keyword, otomatis pada atribut dan method maka akan diwariskan kepada class child. Pada program 4 atribut nip, nama, dan gaji memakai keyword super maka akan langsung merujuk pada atribut di class parent, serta this dipakai pada nidn jadi bisa ditampilkan.

2. Mengapa error terjadi pada program no 6?

Jawab : Karena keyword **super** akan langsung merujuk pada class parent, sedangkan class parent tidak ada atribut nidn.

7. PERCOBAAN 5 (super & overriding)

A. TAHAPAN PERCOBAAN

1. Lakukan modifikasi kembali pada method getAllInfo(). Run program kemudian amatihasilnya

```
public String getAllInfo(){
    String info = getInfo();
    info += "NIDN\t\t\t\t: " + nidn ;

    return info;
}
```

```
Objek dari class Pegawai dibuat
Objek dari class Dosen dibuat
NIP                : 34329837
Nama               : Yansy Ayuningtyas
Gaji               : 3000000.0
NIDN               : 1989432439
PS D:\project_java\project_java_semester3> █
```


2. Lakukan modifikasi kembali pada method `getAllInfo()`. Run program kemudian amatihasilnya

```
public String getAllInfo(){  
    String info = this.getInfo();  
    info += "NIDN\t\t\t: " + nidn ;  
  
    return info;  
}
```

```
Objek dari class Pegawai dibuat  
Objek dari class Dosen dibuat  
NIP                : 34329837  
Nama                : Yansy Ayuningtyas  
Gaji                : 3000000.0  
NIDN                : 1989432439  
PS D:\project_java\project_java_semester3> 
```

3. Lakukan modifikasi kembali pada method `getAllInfo()`. Run program kemudian amatihasilnya

```
public String getAllInfo(){  
    String info = super.getInfo();  
    info += "NIDN\t\t\t: " + nidn ;  
  
    return info;  
}
```

```
Objek dari class Pegawai dibuat  
Objek dari class Dosen dibuat  
NIP                : 34329837  
Nama                : Yansy Ayuningtyas  
Gaji                : 3000000.0  
NIDN                : 1989432439  
PS D:\project_java\project_java_semester3> 
```

4. Tambahkan method `getInfo()` pada class `Dosen` dan modifikasi method `getAllInfo()` sebagai berikut

```

class Dosen extends Pegawai{
    public String nidn;

    public Dosen(){
        System.out.println(x: "Objek dari class Dosen dibuat");
    }

    public String getInfo(){
        return "NIDN\t\t\t: " + this.nidn + "\n";
    }

    public String getAllInfo(){
        String info = super.getInfo();
        info += this.getInfo();

        return info;
    }
}

```

B. PERTANYAAN

1. Apakah ada perbedaan method getInfo() yang diakses pada langkah 1, 2, dan 3?

Jawab : tidak ada, semua menampilkan hasil yang sama.

- Langkah 1 pada subclass Dosen method getAllInfo() sintaks `String info = getInfo();` , sintaks tersebut berarti menginisiasi variable info dengan method yang ada di parent class dan pada parent class ada method getInfo() juga, maka akan langsung diwariskan pada subclass
- Langkah 2
`String info = this.getInfo();` sintaks tersebut akan merujuk pada method yang ada di subclass lalu sintaks `info += "NIDN\t\t\t: " + nidn + "\n";` merujuk pada atribut di subclass sehingga hasilnya akan sama pada Langkah 1
- Langkah 3 sintaks `String info = super.getInfo();` maka yang akan diakses akan langsung merujuk di parent class lalu di bawahnya ditambah sintaks `info += "NIDN\t\t\t: " + nidn + "\n";` maka hasilnya akan sama pada langkah sebelumnya.

2. Apakah ada perbedaan method super.getInfo() dan this.getInfo() yang dipanggil dalam method getAllInfo() pada langkah 4? Jelaskan!

Jawab :

- method super.getInfo() terdapat keyword **super** maka akan merujuk pada method di parent class, yang dipanggil akan langsung menuju pada parent class
- method this.getInfo() keyword **this** , merujuknya hanya di method subclass, maka yang dipanggil hanya tertuju pada subclass

3. Pada method apakah terjadi overriding? Jelaskan!

Jawab : iya, terjadi overriding . Pada kedua class terdapat nama method yang sama, yaitu

getInfo() yang memiliki signature yang sama juga sehingga method getInfo() di kelas subclass akan meng-override method getInfo() di parent classnya.

8. PERCOBAAN 6 (overloading)

A. TAHAPAN PERCOBAAN

1. Tambahkan constructor baru untuk class Dosen sebagai berikut

```
public Dosen(String nip, double gaji, String nidn){  
    System.out.println(x: "Objek dari class Dosen dibuat dengan constructor berparameter");  
}
```

2. Modifikasi class InheritanceDemo untuk menginstansiasi object baru dengan nama dosen2 dengan constructor yang berparameter. Run program kemudian amati hasilnya.

```
public static void main(String[] args) {  
    Dosen dosen2 = new Dosen(nip: "34329837", gaji: 3000000, nidn: "1989432439" );  
    System.out.println(dosen2.getAllInfo());  
}
```

B. PERTANYAAN

1. Bagaimana hasil nilai nip, nama, gaji, dan nidn yang ditampilkan pada langkah 2? Mengapa demikian?

Jawab : hasil NIP, nama, NIDN akan bernilai null, dan untuk gaji bernilai 0 karena tidak ada perintah (keyword) manakah yang harus dirujuk/diakses terhadap class parent maupun subclass

2. Jelaskan apakah constructor tanpa parameter dan constructor class Dosen yang dibuat padalangkah 1 memiliki signature yang sama?

Jawab : Iya beda, karena constructor tanpa parameter dengan berparameter jelas memiliki signature yang beda dari susunan, tipe data, dan jumlah.

3. Konsep apa dalam OOP yang membolehkan suatu class memiliki constructor atau method dengan nama yang sama dan signature yang berbeda pada satu class?

Jawab : Overloading

9. PERCOBAAN 7 (super - constructor)

A. TAHAPAN PERCOBAAN

1. Modifikasi constructor pada class Dosen sebagai berikut. Run program kemudian amatihasilnya.

```
public Dosen(String nip, String nama, double gaji, String nidn){
    this.nip = nip;
    this.nama = nama;
    this.gaji = gaji;
    this.nidn = nidn;
}
```

```
Objek dari class Pegawai dibuat
NIP                : 34329837
Nama               : Yansy Ayuningtyas
Gaji               : 3000000.0
NIDN              : 1989432439

PS D:\project_java>
```

2. Modifikasi constructor pada class Dosen sebagai berikut. Run program kemudian amatihasilnya.

```
public Dosen(String nip, String nama, double gaji, String nidn){
    super.nip = nip;
    super.nama = nama;
    super.gaji = gaji;
    this.nidn = nidn;
}
```

```
Objek dari class Pegawai dibuat
NIP                : 34329837
Nama               : Yansy Ayuningtyas
Gaji               : 3000000.0
NIDN              : 1989432439
PS D:\project_java>
```

3. Modifikasi constructor pada class Dosen sebagai berikut. Run program kemudian amatihasilnya.

```
public Dosen(String nip, String nama, double gaji, String nidn){
    super();
    super.nip = nip;
    super.nama = nama;
    super.gaji = gaji;
    this.nidn = nidn;
}
```

```
Objek dari class Pegawai dibuat
NIP                : 34329837
Nama               : Yansy Ayuningtyas
Gaji               : 3000000.0
NIDN               : 1989432439

PS D:\project_java> 
```

4. Hapus/comment constructor tanpa parameter dari class Pegawai. Tambahkan constructor baru untuk class Pegawai sebagai berikut. Run program kemudian amati hasilnya.

```
public class Pegawai {
    public String nip;
    public String nama;
    public double gaji;

    // public Pegawai(){
    //     System.out.println("Objek dari class Pegawai dibuat");
    // }

    public Pegawai(String nip, String nama, double gaji){
        this.nip = nip;
        this.nama = nama;
        this.gaji = gaji;
    }

    public String getInfo(){
        String info = "";
        info += "NIP\t\t\t\t\t: " + nip + "\n";
        info += "Nama\t\t\t\t\t: " + nama + "\n";
        info += "Gaji\t\t\t\t\t: " + gaji + "\n";

        return info;
    }
}
```

5. Modifikasi constructor pada class Dosen sebagai berikut. Run program kemudian amatihasilnya.

```
public Dosen(String nip, String nama, double gaji, String nidn){
    this.nidn = nidn;
    super(nip, nama, gaji);
}
```

```
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
Implicit super constructor Pegawai() is undefined. Must explicitly invoke another constructor
Constructor call must be the first statement in a constructor

at pemrogramanberbasisobjek.pertemuan7.Dosen.<init>(Pegawai.java:31)
at pemrogramanberbasisobjek.pertemuan7.InheritanceDemo.main(InheritanceDemo.java:15)PS D:\project_java> 
```

6. Modifikasi constructor pada class Dosen sebagai berikut. Run program kemudian amati hasilnya.

```
public Dosen(String nip, String nama, double gaji, String nidn){
    super(nip, nama, gaji);
    this.nidn = nidn;
}
```

```
NIP                : 34329837
Nama               : Yansy Ayuningtyas
Gaji               : 3000000.0
NIDN               : 1989432439

PS D:\project_java> 
```

B. PERTANYAAN

1. Apakah terdapat perbedaan hasil pada langkah 1 dan 2? Jelaskan!

Jawab : Tidak ada, semua menampilkan hasil yang sama

- Pada Langkah 1 menggunakan keyword **this, this** hanya akan langsung merujuk pada subclass sedang subclass otomatis mewarisi atribut dan method dari parent classnya
- Pada Langkah 2 menggunakan keyword **super, super** akan langsung merujuk pada parent class kemudian parent class meng-inherit atribut dan method terhadap subclassnya

Dengan demikian akan menampilkan hasil yang sama dari Langkah 1 dan 2.

2. Apakah terdapat perbedaan hasil pada langkah 2 dan 3? Jelaskan!

Jawab : Tidak, semua menampilkan hasil yang sama karena semua menggunakan keyword **super** perbedaanya pada Langkah 3 ada pemanggilan method constructor yang merujuk pada constructor parent class, yaitu **super();**

3. Mengapa terjadi error pada langkah 4?

Jawab : karena dalam pemanggilan constructor lain harus secara eksplisit sedangkan constructor pada class Pegawai (super constructor) belum ditentukan atau masih secara implisit.

4. Apa perbedaan **super()** yang dipanggil pada langkah 3 dan 6?

Jawab : pada Langkah 3 keyword **super**, yaitu sintaks **super();** tidak ada argumennya, masing-masing atribut dideklarasikan menggunakan keyword **super** untuk dapat merujuk pada parent classnya, kecuali atribut **nidn**. Sedangkan pada Langkah 6 **super();** diisikan argument untuk langsung mengakses atribut terhadap parent classnya menjadi **super(nip, nama, gaji);** .

5. Mengapa terjadi error pada langkah 5?

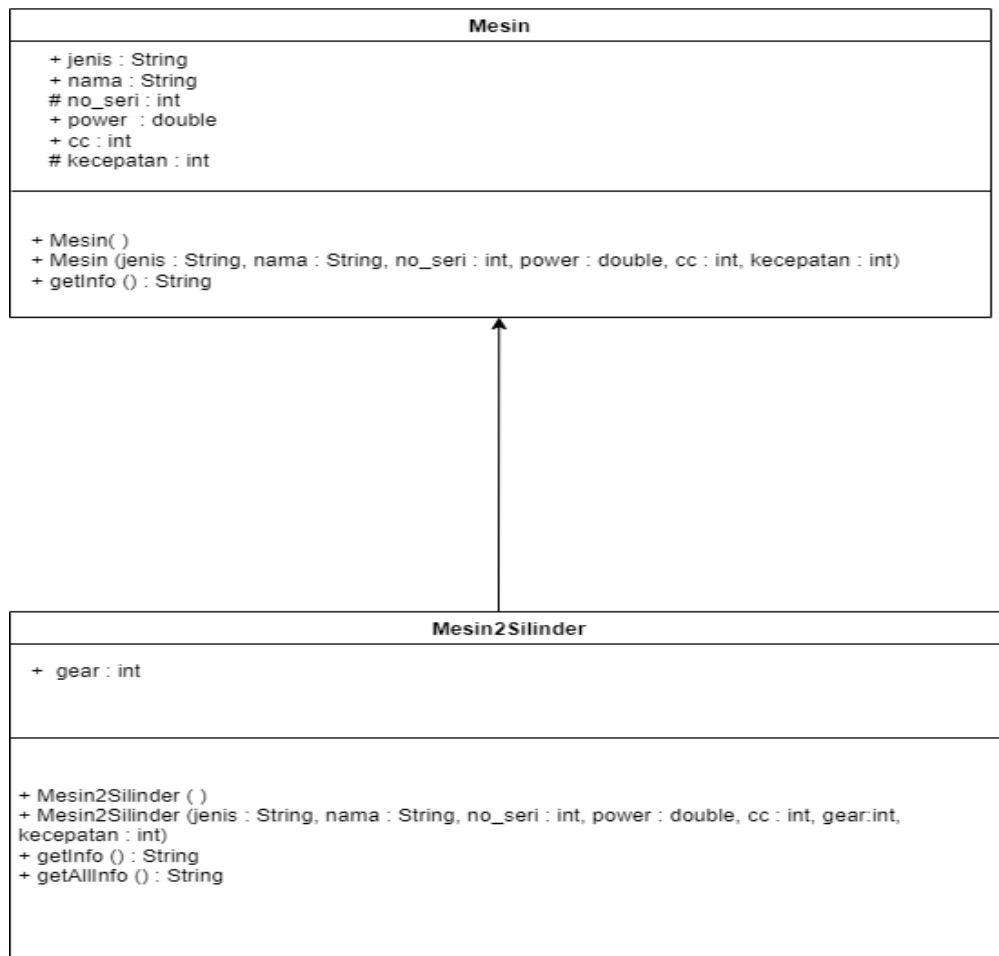
Jawab : urutan pemanggilan constructor dari parent class harus dilakukan pada statement pertama di sebuah constructor subclass

10. TUGAS

1. Tentukan sebuah class yang merupakan turunan dari class yang lain.
2. Buat 3 atribut pada parent class kemudian tambahkan minimal 1 atribut pada child class.
3. Lakukan method overloading dengan membuat 2 constructor yaitu constructor tanpa parameter dan constructor berparameter pada masing-masing class. Panggil constructor `super()` berparameter untuk membuat object dari parent class pada constructor child class.
4. Lakukan method overriding dengan membuat method dengan nama dan signature yang sama pada parent class dan child class.
5. Lakukan instansiasi objek child class pada main class kemudian print info nya.

Diagram class :

Single Inheritance



Code java

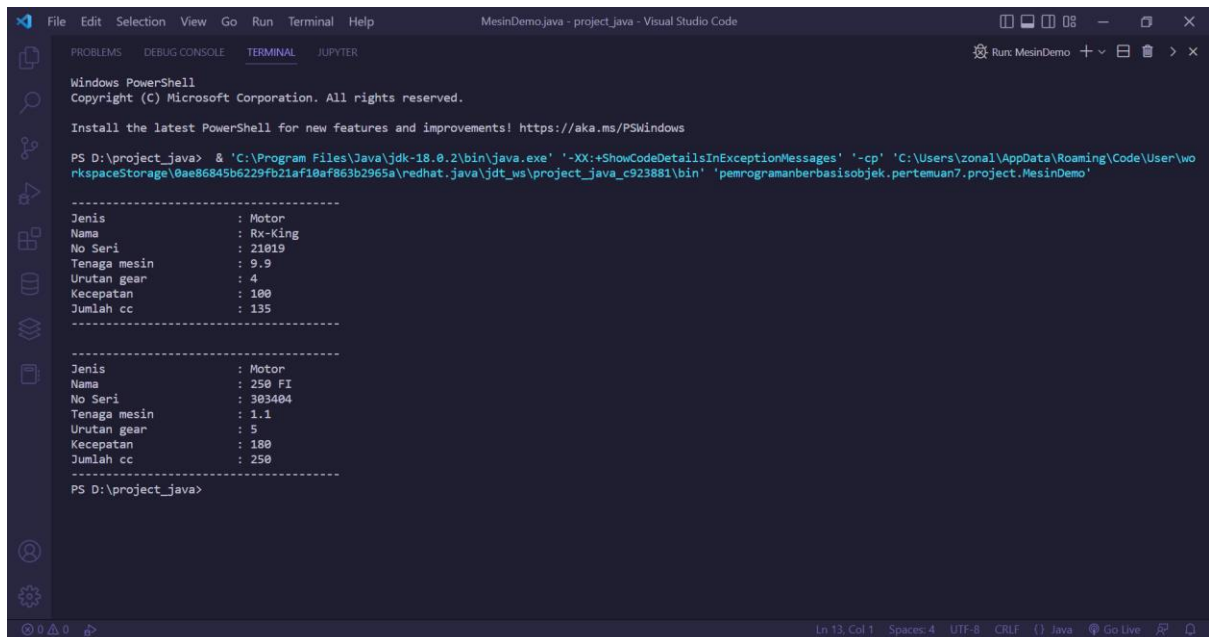
1. class Mesin

```
File Edit Selection View Go Run Terminal Help Mesin.java - project.java - Visual Studio Code
diagram_mesin.drawio Mesin.java MesinDemo.java
project_java_semester3 > src > pemrogramanberbasisobjek > pertemuan7 > project > Mesin.java > Mesin
1 package pemrogramanberbasisobjek.pertemuan7.project;
2
3 public class Mesin {
4     public String jenis;
5     public String nama;
6     protected int no_seri;
7     public double power;
8     public int cc;
9     protected int kecepatan;
10
11     public Mesin(){
12         System.out.println("Ini adalah mesin utama");
13     }
14
15     public Mesin(String jenis, String nama, int no_seri, double power, int cc, int kecepatan){
16         this.jenis = jenis;
17         this.nama = nama;
18         this.no_seri = no_seri;
19         this.power = power;
20         this.cc = cc;
21         this.kecepatan = kecepatan;
22     }
23
24     public String getInfo(){
25         String info = "";
26         info += "\n-----\n";
27         info += "Jenis \t\t\t: " + this.jenis + "\n";
28         info += "Nama \t\t\t: " + this.nama + "\n";
29         info += "No Seri \t\t: " + this.no_seri + "\n";
30         info += "Tenaga mesin\t\t: " + this.power + "\n";
31
32         return info;
33     }
34 }
35
36 class Mesin2Silinder extends Mesin{
37     public int gear;
38
39     public Mesin2Silinder(){
40         System.out.println("Ini adalah mesin 2 silinder");
41     }
42
43     public Mesin2Silinder(String jenis, String nama, int no_seri, double power, int cc, int gear, int kecepatan){
44         super(jenis, nama, no_seri, power, cc, kecepatan); //memanggil super() berparameter untuk membuat object parent class pada constructor class
45         this.gear = gear;
46     }
47
48     public String getInfo(){
49
50         String ability = "";
51         ability += "Urutan gear\t\t: " + this.gear + "\n";
52         ability += "Kecepatan\t\t: " + this.kecepatan + "\n";
53         ability += "Jumlah cc\t\t: " + this.cc;
54         ability += "\n-----";
55
56         return ability;
57     }
58
59     public String getAllInfo(){
60         String info = super.getInfo();
61         info += this.getInfo();
62
63         return info;
64     }
65 }
66
```

2. class MesinDemo

```
File Edit Selection View Go Run Terminal Help MesinDemo.java - project.java - Visual Studio Code
diagram_mesin.drawio Mesin.java MesinDemo.java
project_java_semester3 > src > pemrogramanberbasisobjek > pertemuan7 > project > MesinDemo.java > ...
1 package pemrogramanberbasisobjek.pertemuan7.project;
2
3 public class MesinDemo {
4     public static void main(String[] args) {
5         Mesin2Silinder mesin2s = new Mesin2Silinder(jenis "Motor", nama "Rx-King", no_seri 21019, power 9.9, cc 135, gear 4, kecepatan 180);
6         Mesin2Silinder mesin3s = new Mesin2Silinder(jenis "Motor", nama "250 FI", no_seri 303404, power 1.1, cc 250, gear 5, kecepatan 180);
7
8         System.out.println(mesin2s.getAllInfo());
9         System.out.println(mesin3s.getAllInfo());
10    }
11 }
```


3. Hasil running



```
File Edit Selection View Go Run Terminal Help MesinDemo.java - project_java - Visual Studio Code
PROBLEMS DEBUG CONSOLE TERMINAL JUPYTER
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\project_java> & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\zonal\AppData\Roaming\Code\User\workspacesStorage\0ae86845b6229fb21af10af863b2965a\redhat.java\jdt_ws\project_java_c923881\bin' 'pemrogramanberbasisobjek.pertemuan7.project.MesinDemo'

-----
Jenis      : Motor
Nama       : Rx-King
No Seri    : 21019
Tenaga mesin : 9.9
Urutan gear : 4
Kecepatan  : 100
Jumlah cc  : 135
-----

Jenis      : Motor
Nama       : 250 FI
No Seri    : 303404
Tenaga mesin : 1.1
Urutan gear : 5
Kecepatan  : 180
Jumlah cc  : 250
-----
PS D:\project_java>
```