



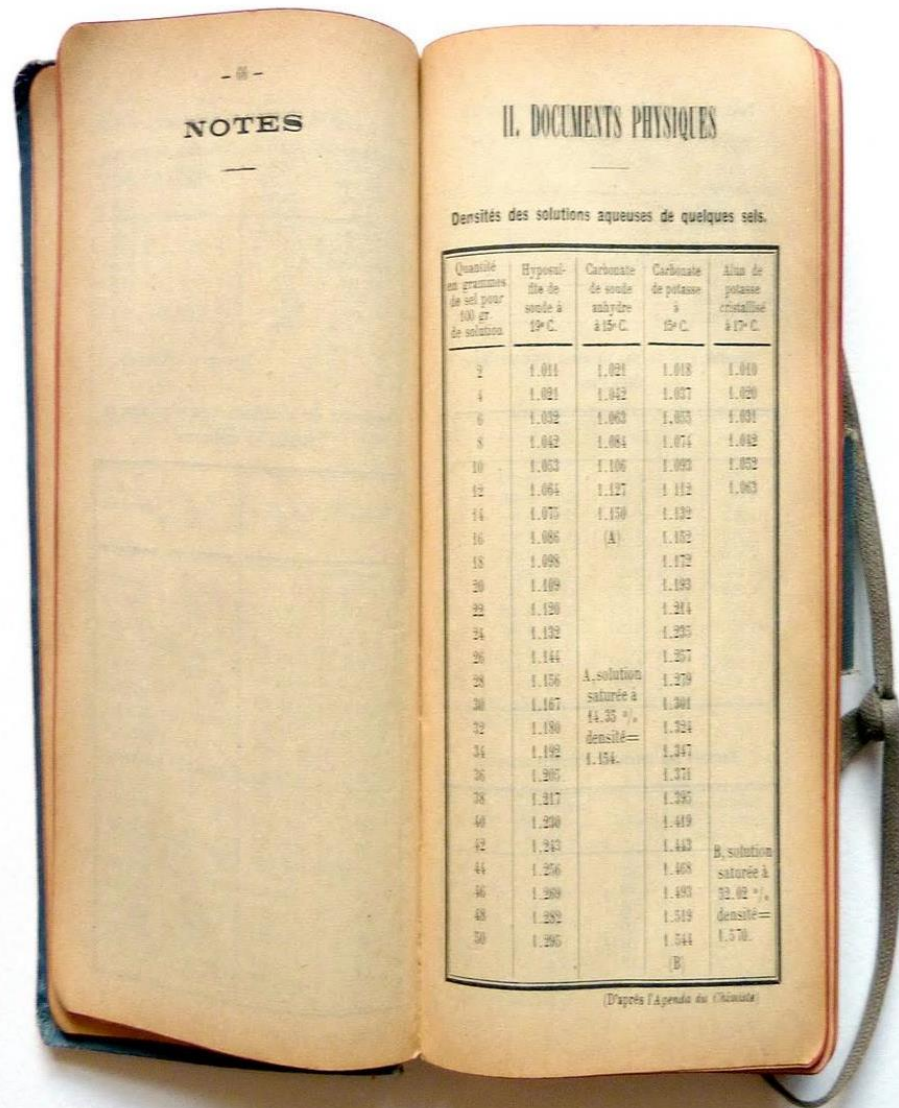
# Pemrograman Berorientasi Objek



## Pertemuan 6

## Pewarisan (Inheritance)

Pemateri : Chrismikha Hardyanto S.Kom., M.Kom.



# KONTEN PERKULIAHAN

- Konsep Dasar Pewarisan
- Membuat pewarisan pada JAVA
- Kata kunci Protected, Super, & Final
- Contoh Kasus Pewarisan pada program JAVA

Setelah kita memahami bagaimana membuat CLASS dan OBJECT didalam program, selanjutnya kita lihat konsep PBO lainnya yaitu **Inheritance**

# Konsep Dasar Pewarisan

- ❑ **Pewarisan (Inheritance)** Merupakan salah satu prinsip dasar PBO yang memungkinkan sebuah **kelas** untuk dapat **mempunyai/memiliki seluruh isi member (atribut & method)** dari **kelas lainnya**.
- ❑ **Artinya,** Isi suatu class (atribut & method) dapat **diturunkan** kepada **Class yang menjadi turunan** class tersebut.
- ❑ Class yang menurunkan disebut **Parent Class / Super Class**. Sedangkan Class yang menjadi turunan disebut **Child Class / Sub Class**

{OOP}

# Ilustrasi Pewarisan

- ❑ Saat sebuah class **diturunkan** ke kelas lain (sebagai parent class), maka **semua atribut dan method** yang ada di class parent secara otomatis **dimiliki oleh class child** (tampa perlu dideklarasikan kembali)
- ❑ Dalam konsep **pewarisan di JAVA**, class child hanya bisa punya **satu class parent**. sedangkan class parent boleh mempunyai **lebih dari satu child**

Pegawai
-nama: String -alamat: String -noTelpon: String
+tampilStatus(): void

[SUPER CLASS]

DosenTetap
-nip: String

[SUB CLASS]

DosenLayanan

[SUB CLASS]

# Manfaat Pewarisan

## ❑ REUSABILITY

Mengurangi **penulisan kode program** dengan tujuan yang sama **berulang kali** pada class - class didalam program. Kita bisa menggunakan kembali kode program yang terdapat pada class parent (super class) melalui class child (sub class)

## ❑ GENERALISASI

**Menyederhanakan** isi/member dari suatu class dengan **membaginya kedalam class - class yang lebih spesifik** dalam melakukan tujuan tertentu didalam program. Kode didalam suatu class akan lebih mudah untuk dikelola oleh programmer

{OOP}

# Membuat Pewarisan Pada JAVA

- ❑ Untuk menerapkan konsep pewarisan didalam program JAVA, kita dapat menggunakan **kata kunci extends** pada deklarasi dari **class child (sub class) nya**

```
<hak akses> class <SubClass> extends <SuperClass> {  
    //Isi dari SubClass (Atribut dan atau Method)  
}
```

- ❑ Contoh Penulisan pada JAVA :

```
public class Mobil extends Kendaraan {  
    ...  
}
```

## JAVA

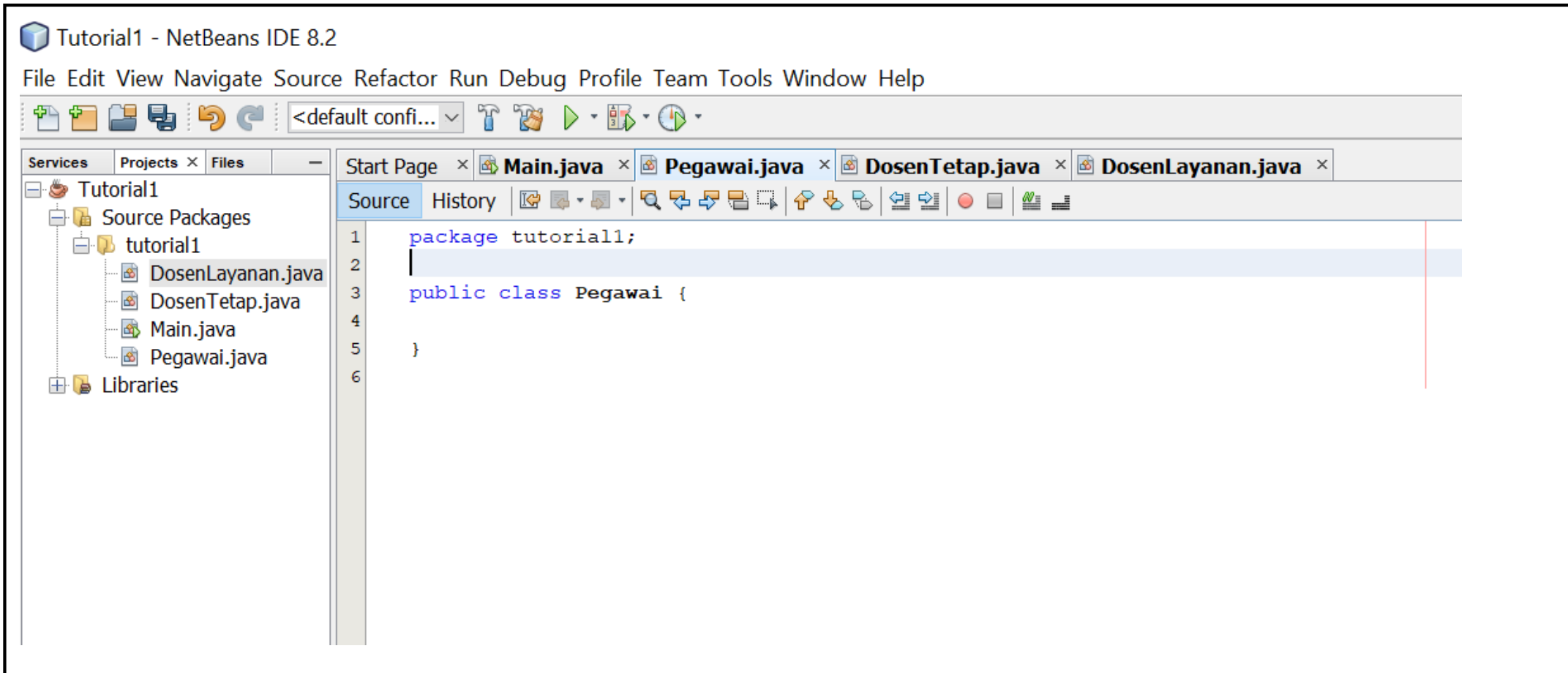
```
public class A {  
    ...  
}
```

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

# Contoh Pewarisan Pada JAVA

Berikut adalah contoh Coding membuat pewarisan menggunakan JAVA dengan Netbeans.

- ❑ **Langkah 1 :** Buatlah 3 buah class dengan nama Pegawai, DosenTetap, & Dosen Layanan





# Contoh Pewarisan Pada JAVA (2)

- ❑ **Langkah 2 :** Isilah Class Pegawai dengan atribut dan method berikut

```
1  package tutorial1;  
2  
3  public class Pegawai {  
4  
5      //Deklarasi Atribut pada class Pegawai  
6      private String nama;  
7      private String alamat;  
8      private String noTelpon;  
9  
10     //Deklarasi Getter & Setter untuk atribut Private  
11     public String getNama() { ...3 lines }  
14  
15     public void setNama(String nama) { ...3 lines }  
18  
19     public String getAlamat() { ...3 lines }  
22  
23     public void setAlamat(String alamat) { ...3 lines }  
26  
27     public String getNoTelpon() { ...3 lines }  
30  
31     public void setNoTelpon(String noTelpon) { ...3 lines }  
34  
35     //Deklarasi Method pada Class Pegawai  
36     public void tampilStatus() {  
37         System.out.println(this.nama+" adalah PEGAWAI di UNIKOM");  
38     }  
39 }
```

# Contoh Pewarisan Pada JAVA (3)

- ❑ **Langkah 3** : Buatlah Class DosenTetap yang menjadi **child** dari class Pegawai

```
1  package tutorial1;  
2  
3  public class DosenTetap extends Pegawai {  
4  
5      //Deklarasi Atributdari Class DosenTetap  
6      private String nip;  
7  
8      //Deklarasi Getter & Setter  
9      public String getNip() { ...3 lines }  
12  
13      public void setNip(String nip) { ...3 lines }  
16  }  
17
```

\*Dengan **kata kunci extends** , artinya class DosenTetap merupakan turunan (subclass/child class) dari pegawai sehingga isi class pegawai bisa dipanggil dari object DosenTetap

# Contoh Pewarisan Pada JAVA (3)

- ❑ **Langkah 4** : Buatlah Class DosenLayanan yang menjadi **child** dari class **Pegawai**

```
1  package tutorial1;  
2  
3  public class DosenLayanan extends Pegawai {  
4  
5      //Untuk kasus ini kosongkan saja isi Class nya  
6  
7  }  
8  
9  
10
```

\*Dengan **kata kunci extends** , artinya class DosenLayanan merupakan turunan (subclass/child class) dari pegawai sehingga isi class pegawai bisa dipanggil dari object DosenLayanan

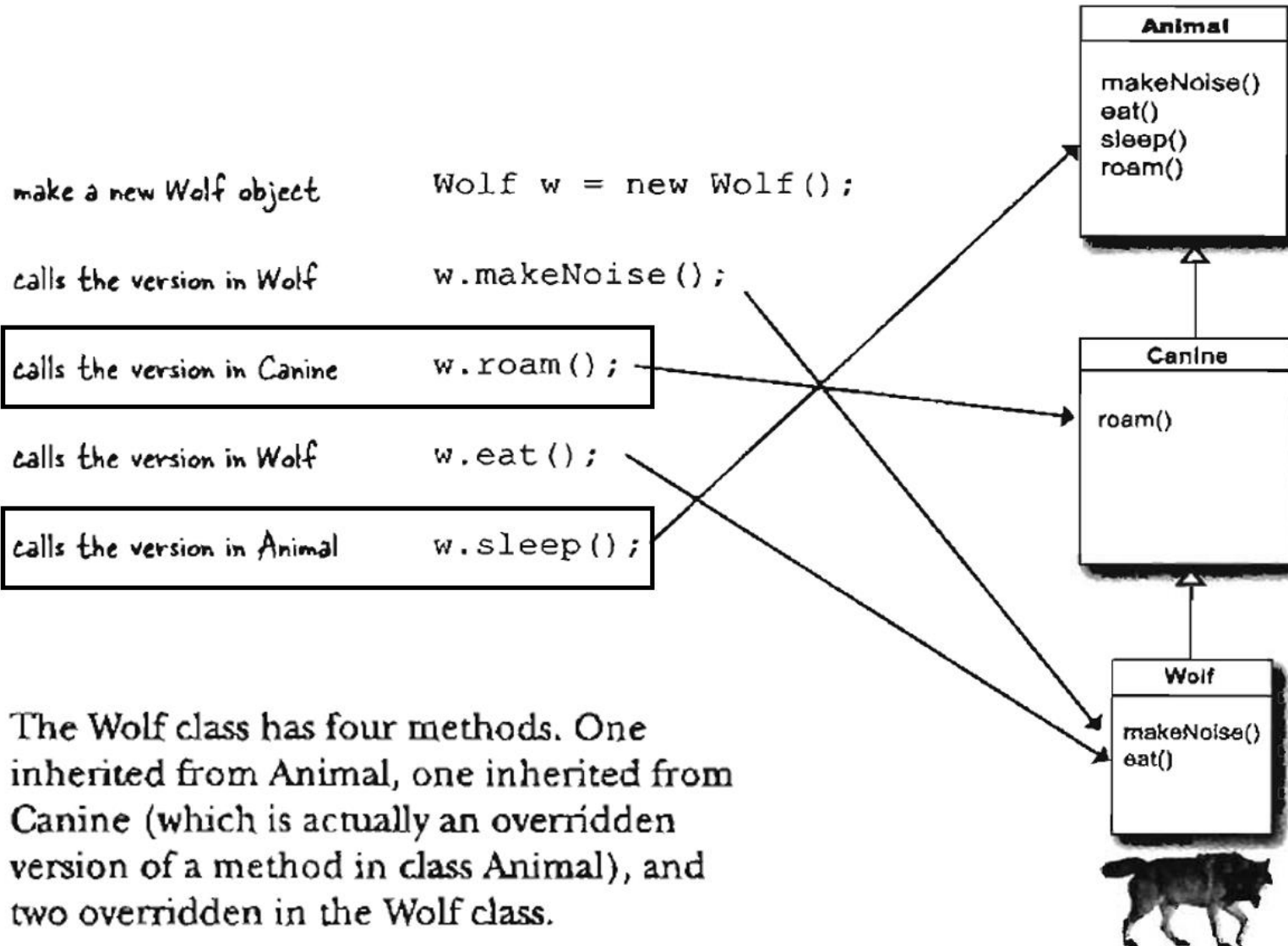
# Contoh Pewarisan Pada JAVA (4)

- ❑ **Langkah 5 :** Implementasikan kode pada class DosenTetap & DosenLayanan pada Class Main

```
1 package tutorial1;  
2  
3 public class Main {  
4  
5     public static void main(String[] args) {  
6         /*Deklarasi Object dari Class DosenTetap & DosenLayanan  
7         Isilah nama, alamat, & noTelpn kemudian tampilkan ke layar*/  
8         DosenTetap pegawai1 = new DosenTetap();  
9         DosenLayanan pegawai2 = new DosenLayanan();  
10  
11         //Data Pegawai ke-1  
12         pegawai1.setNama("Chrismikha Hardyanto");  
13         pegawai1.setNip("41277006141");  
14         pegawai1.setAlamat("Margahayu");  
15         pegawai1.setNoTelpon("0819xxx");  
16         System.out.println("Data Pegawai UNIKOM ke-1");  
17         System.out.println("Nama Pegawai      : " +pegawai1.getNama());  
18         System.out.println("NIP Pegawai       : " +pegawai1.getNip());  
19         System.out.println("Alamat          : " +pegawai1.getAlamat());  
20         System.out.println("No Telepon       : " +pegawai1.getNoTelpon());  
21         pegawai1.tampilStatus();  
22         System.out.println("");  
23  
24         //Data Pegawai ke-2  
25         System.out.println("Data Pegawai UNIKOM ke-2");  
26         pegawai2.setNama("Rizki Adam Kurniawan");  
27         pegawai2.setAlamat("Cibaduyut");  
28         pegawai2.setNoTelpon("0818xxx");  
29         System.out.println("Nama Pegawai      : " +pegawai2.getNama());  
30         System.out.println("Alamat           : " +pegawai2.getAlamat());  
31         System.out.println("No Telepon       : " +pegawai2.getNoTelpon());  
32         pegawai2.tampilStatus();  
33     }  
34 }
```

Perhatikan object dari DosenTetap & DosenLayanan tetap bisa menggunakan atribut nama,alamat, & noTelpon walaupun tidak dideklarasikan pada class nya (itulah pewarisan)

# Mengakses isi Class Dari Object



- ❑ Dari contoh sebelumnya dapat dilihat jika kita menggunakan **object dari class turunan** untuk mengakses isi classnya, maka secara otomatis kita bisa **mengakses seluruh isi** class parentnya
- ❑ Sebaliknya jika kita membuat **object dari parent class**, maka **tidak dapat** digunakan untuk mengakses isi **child class nya**

# Kata Kunci

# Protected, Super, & Final

**\*Beberapa kata kunci umum di JAVA yang dapat dimanfaatkan didalam konsep pewarisan**

# Kata Kunci Protected

- ❑ **Kata kunci protected** dapat digunakan jika kita ingin memberikan **tingkat akses protected** pada member dari class parent
- ❑ Untuk menambahkan kata kunci protected, Kita tinggal **mendeklarasikan di posisi paling awal** dari atribut atau method didalam class (Sama seperti cara deklarasi untuk hak akses lainnya)
- ❑ Contoh Penulisan didalam JAVA :

```
protected String nama;  
protected String alamat;
```

{OOP}

# Contoh Kata Kunci Protected

- ❑ Pada program sebelumnya, mari Kita rubah hak akses pada atribut nama, alamat, & noTelpon

```
1  package tutorial1;  
2  
3  @  
4  public class Pegawai {  
5      //Deklarasi Atribut pada class Pegawai  
6      protected String nama;  
7      protected String alamat;  
8      protected String noTelpon;  
9  
10     //Deklarasi Getter & Setter untuk atribut Private  
11     public String getNama() {...3 lines }  
14  
15     public void setNama(String nama) {...3 lines }  
18  
19     public String getAlamat() {...3 lines }  
22  
23     public void setAlamat(String alamat) {...3 lines }  
26  
27     public String getNoTelpon() {...3 lines }  
30  
31     public void setNoTelpon(String noTelpon) {...3 lines }  
34  
35     //Deklarasi Method pada Class Pegawai  
36     public void tampilStatus() {  
37         System.out.println(this.nama+" adalah PEGAWAI di UNIKOM");  
38     }  
39 }
```



# Kata Kunci Super

- ❑ **Kata kunci Super** digunakan untuk **mengakses parent class** (Bisa atribut, method, ataupun konstruktor) pada child class nya.
- ❑ Kata kunci super **mirip** dengan kata kunci this, Bedanya kata kunci super adalah **objek yang mewakili/mengacu class parent (super class)**
- ❑ Pemanfaatan kata kunci super adalah untuk menghindari terjadinya **variable shadowing** pada child class dan juga digunakan untuk menurunkan konstruktor dari parent class ke child class nya

{OOP}

# Pemanfaatan Kata Kunci Super

1

Super can be used to refer immediate parent class instance variable.

2

Super can be used to invoke immediate parent class method.

3

super() can be used to invoke immediate parent class constructor.

{OOP}

# Menurunkan Konstruktor

- ❑ Apabila kita tidak membuat konstruktor apapun pada parent class (konstruktor default) , Maka kita **tidak wajib untuk memanggil konstruktor** pada child class.
- ❑ Namun jika ada **konstruktor berparameter** pada **parent class** maka kita **wajib untuk memanggilnya** kembali didalam child class nya (dengan jumlah & tipe parameter yang sama).
- ❑ Jika tidak dipanggil kembali, maka program JAVA akan **error.**

{OOP}

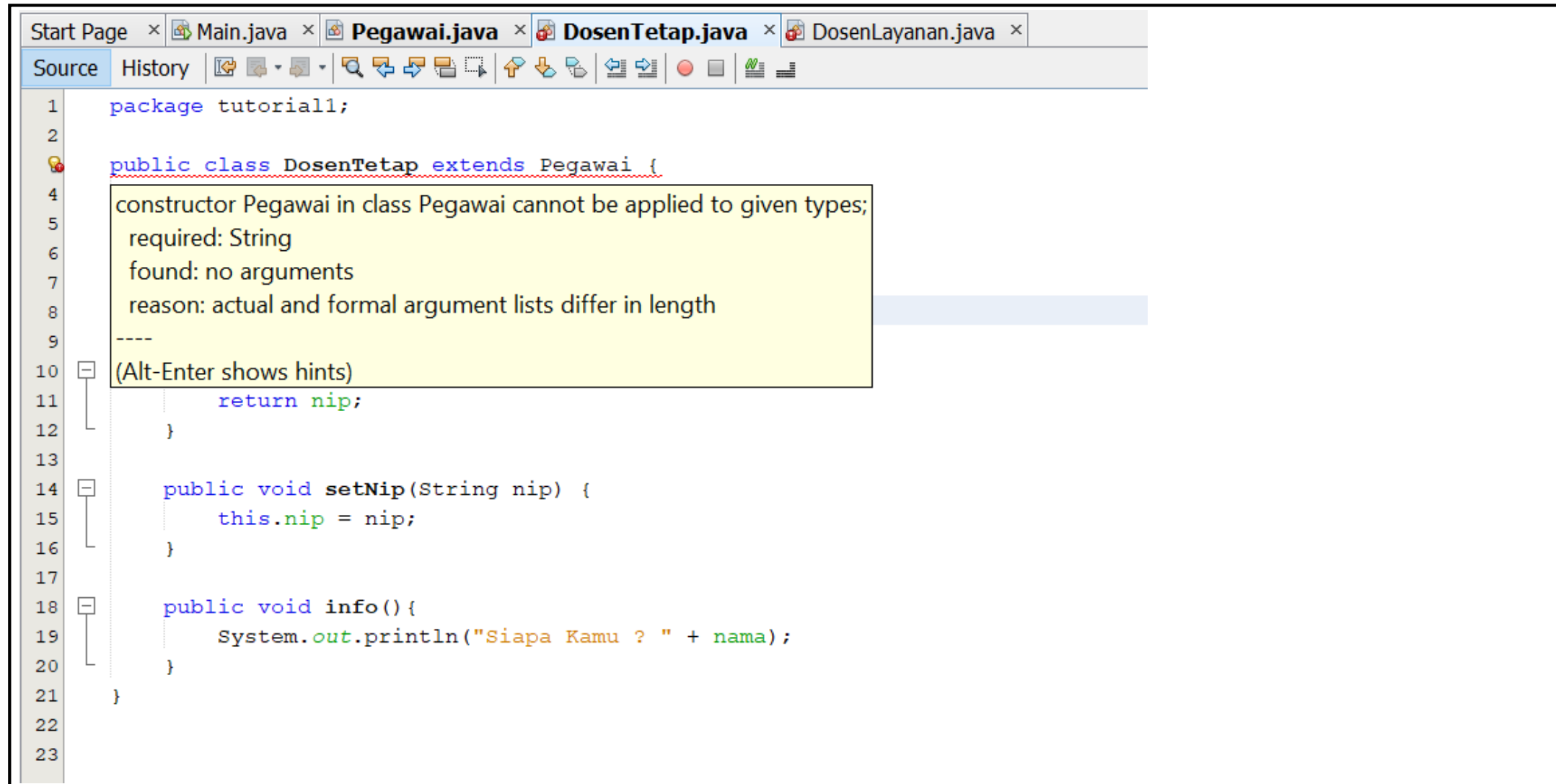
# Contoh Menurunkan Konstruktor

- ❑ **Langkah 1 :** Buatlah sebuah konstruktor pada class Pegawai dengan 1 parameter String

```
1  package tutorial1;  
2  
3  @  
4  public class Pegawai {  
5      //Deklarasi Atribut pada class Pegawai  
6      protected String nama;  
7      protected String alamat;  
8      protected String noTelpon;  
9  
10     //Deklarasi Konstruktor dengan 1 parameter String  
11     public Pegawai(String nama) {  
12         this.nama = nama;  
13     }  
14  
15     //Deklarasi Getter & Setter untuk atribut Private  
16     public String getNama() {  
17         return nama;  
18     }  
19  
20     public void setNama(String nama) { ...3 lines }  
23  
24     public String getAlamat() { ...3 lines }  
27  
28     public void setAlamat(String alamat) { ...3 lines }  
31  
32     public String getNoTelpon() { ...3 lines }  
35  
36     public void setNoTelpon(String noTelpon) { ...3 lines }  
39  
40     //Deklarasi Method pada Class Pegawai  
41     public void tampilStatus() {  
42         System.out.println(this.nama+" adalah PEGAWAI di UNIKOM");  
43     }  
44 }
```

# Contoh Menurunkan Konstruktor(2)

- ❑ **Langkah 2 :** Pindah ke class DosenTetap & Dosen Layanan, aan muncul error



```
Start Page x Main.java x Pegawai.java x DosenTetap.java x DosenLayanan.java x
Source History
1 package tutorial1;
2
3
4 public class DosenTetap extends Pegawai {
5     constructor Pegawai in class Pegawai cannot be applied to given types;
6     required: String
7     found: no arguments
8     reason: actual and formal argument lists differ in length
9     ----
10    (Alt-Enter shows hints)
11        return nip;
12    }
13
14    public void setNip(String nip) {
15        this.nip = nip;
16    }
17
18    public void info() {
19        System.out.println("Siapa Kamu ? " + nama);
20    }
21 }
22
23
```

# Contoh Menurunkan Konstruktor(3)

- ❑ **Langkah 3** : Untuk menurunkan konstruktor, buatlah konstruktor **sejenis** di class child

```
1  package tutorial11;
2
3  public class DosenTetap extends Pegawai {
4
5      //Deklarasi Atribut dari Class DosenTetap
6      private String nip;
7
8      public DosenTetap(String nama) {
9          //Gunakan Kata kunci super untuk memanggil Konstruktor di class parent
10         super(nama);
11     }
12
13     //Deklarasi Getter & Setter
14     public String getNip() {...3 lines }
15
16
17     public void setNip(String nip) {...3 lines }
18
19
20
21
22     public void info() {...3 lines }
23
24
25 }
```

\*Menurunkan constructor pada class child caranya adalah **memanggil constructor dari class parent** dengan menggunakan **kata kunci super**. Format penulisannya adalah **super (parameter)**

# Contoh Menurunkan Konstruktor(4)

- ❑ **Langkah 4 :** Lakukan hal yang sama pada child class lainnya

```
1  package tutorial1;  
2  
3  public class DosenLayanan extends Pegawai {  
4  
5      public DosenLayanan(String nama){  
6          //Gunakan Kata kunci super untuk memanggil Konstruktor di class parent  
7          super(nama);  
8      }  
9      //Untuk kasus ini kosongkan saja isi Class nya  
10  
11 }  
12  
13
```

\*Menurunkan constructor pada class child caranya adalah **memanggil constructor dari class parent** dengan menggunakan **kata kunci super**. Format penulisannya adalah **super (parameter)**

# Kata Kunci Final

- ❑ **Kata kunci final** dapat digunakan sebagai penanda bahwa suatu isi class **tidak dapat diubah/diganti**. kata kunci final dapat diimplementasikan pada **atribut, method , dan kelas** itu sendiri.
- ❑ **Final Atribut**, artinya isi dari atribut tersebut tidak dapat diganti (konstanta).
- ❑ **Final Method**, artinya method tersebut kodenya tidak dapat di override oleh sub classnya
- ❑ **Kelas final**, artinya kelas tersebut tidak bisa diturunkan (Menjadi parent class/super class)

{OOP}



# Contoh Kata kunci Final pada Class

- ❑ Coba tambahkan **kata kunci final** pada **class Pegawai**, Lihat yang terjadi pada program

```
1 package tutorial1;  
2  
3 public final class Pegawai {  
4  
5     //Deklarasi Atribut pada class Pegawai  
6     protected String nama;  
7     protected String alamat;  
8     protected String noTelpon;  
9  
10    //Deklarasi Konstruktor dengan 1 parameter String  
11    public Pegawai(String nama){  
12        this.nama = nama;  
13    }  
14  
15    //Deklarasi Getter & Setter untuk atribut Private  
16    public String getNama() {  
17        return nama;  
18    }  
19  
20    public void setNama(String nama) {...3 lines }  
21  
22    public String getAlamat() {...3 lines }  
23  
24    public void setAlamat(String alamat) {...3 lines }  
25  
26    public String getNoTelpon() {...3 lines }  
27  
28    public void setNoTelpon(String noTelpon) {...3 lines }  
29  
30    //Deklarasi Method pada Class Pegawai  
31    public void tampilStatus(){  
32        System.out.println(this.nama+" adalah PEGAWAI di UNIKOM");  
33    }  
34 }  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44
```

```
1 package tutorial1;  
2  
3 public class DosenTetap extends Pegawai {  
4  
5     cannot inherit from final Pegawai  
6     -----  
7     (Alt-Enter shows hints)  
8  
9     public DosenTetap(String nama){  
10        //Gunakan Kata kunci super untuk memanggil Konstruktor  
11        super(nama);  
12    }  
13  
14    //Deklarasi Getter & Setter  
15    public String getNip() {...3 lines }  
16  
17    public void setNip(String nip) {...3 lines }  
18  
19    public void info(){...3 lines }  
20 }  
21  
22  
23  
24  
25  
26
```

\*Dapat dilihat jika kita mencoba menurunkan class final ke class lainnya, maka program akan **ERROR**

Setelah memahami **Materi Hari ini** pada JAVA,  
mari kita berlatih

**\*Silakan buka modul praktikum pertemuan 6&7**

# Terima Kasih