18

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN KE-8

****

**Disusun oleh :**

**NAMA : FERDI DIRGANTARA**

**NIM : 175410039**

**JURUSAN : TEKNIK INFORMATIKA**

**JENJANG : S1**

LABORATORIUM TERPADU

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTERAKAKOMYOGYAKARTA

2019

1. **TEORI SINGKAT**

Dalam Java, juga memungkinkan untuk mendeklarasikan class-class yang tidak dapat menjadi subclass. Class ini dinamakan **class final**. Untuk mendeklarasikan class untuk menjadi final kita hanya menambahkan kata kunci **final** dalam deklarasi class. Sebagai contoh, jika kita ingin class Person untuk dideklarasikan final, kita tulis,

public **final** class Person

{

//area kode

}

Beberapa class dalam Java API dideklarasikan secara final untuk memastikan sifatnya tidak dapat di-*override*. Contoh-contoh dari class ini adalah Integer, Double, dan String.

Ini memungkinkan dalam Java membuat method yang tidak dapat di-override. Method ini dapat kita panggil method final. Untuk mendeklarasikan method untuk menjadi final, kita tambahkan kata kunci final ke dalam deklarasi method. Contohnya, jika kita ingin method getName dalam class Person untuk dideklarasikan final, kita tulis,

public final String getName(){

return name;

}

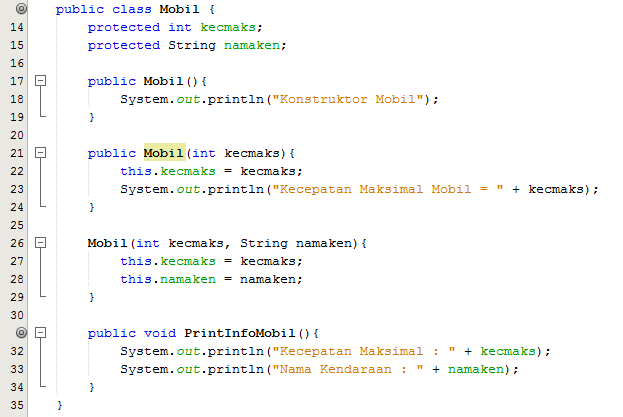
Method static juga secara otomatis final. Ini artinya Anda tidak dapat membuatnya override.

**Keyword super**

Pada saat penciptaan objek dari *subclass* (kelas anak), konstruktor dari  
*superclass* (kelas induk) akan dipanggil.

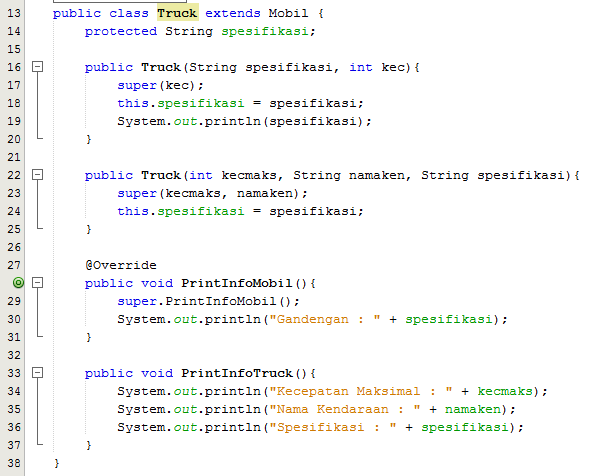
1. **PEMBAHASAN LISTING**

Praktik 1



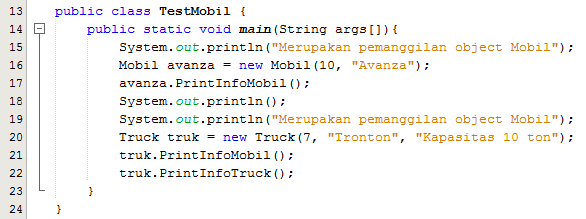
Kelas Mobil akan menerima penyimpanan data untuk sebuah mobil dimana dengan atribut berupa kecepatan maksimal dan nama kendaraan. Kelas Mobil tersebut memiliki tiga buah konstruktor dengan parameter yang berbeda-beda yaitu konstruktor default (tanpa parameter), konstruktor dengan parameter kecepatan maksimal kendaraan dan konstruktor dengan parameter kecepatan maksimal serta nama kendaraan. Selain konstruktor terdapat method PrintInfoMobil() yang digunakan untuk menampilkan data yang telah disimpan pada attribute *kecmaks* dan *namaken* diatas. Kelas ini akan digunakan sebagai kelas induk yang nantinya akan menjadi pewaris untuk kelas lainnya.

Praktik 2



Kelas Truck diatas merupakan subclass dari kelas Mobil (Pewarisan) dimana kelas Mobil yang sebelumnya dibuat dapat diakses melalui kelas Truck diatas dengan memanfaatkan keyword super juga untuk mengakses attribute, konstruktor, maupun method dari kelas induk (kelas Mobil). Pada kelas Truck diatas terdapat konstruktor untuk kelas truck tersebut yang mirip seperti pada kelas Mobil namun ditambahkan attribute spesifikasi dan juga terdapat method override pada kelas diatas untuk memodifikasi method PrintInfoMobil yang ada pada kelas Mobil sebelumnya serta ditambahkan pula method PrintInfoTruck() untuk menampilkan kecepatan maksimal, nama kendaraan, dan spesifikasi kendaraan.

Praktik 3



Kelas TestMobil diatas digunakan untuk menguji kedua kelas yang telah dibuat serta menambahkan data pula pada masing-masing kelas dengan menggunakan operator *new* pada saat instansiasi seperti yang ditunjukkan pada baris 16 dan baris 20.

Data yang ditambahkan kemudian ditampilkan dengan menggunakan method PrintInfoMobil() dan method PrintInfoTruck().

Praktik 4

**Jelaskan tentang override dan adakah contoh implementasi dari praktik diatas**

Override adalah suatu pemanggilan yang ada pada superclass yang kemudian dapat dimodifikasi lagi. Dari praktik diatas dapat dilihat contohnya ada pada kelas Truck pada method PrintInfoMobil().

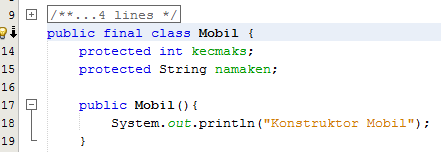
Praktik 5

**Jelaskan tentang penggunaan klausa super dan adakah contoh implementasi dari praktik diatas**

Penggunaan klausa super digunakan untuk mengakses kelas induk dari suatu subclass seperti yang dicontohkan pada kelas Truck diatas yaitu super(kec), super(kecmaks, namaken), dan super.PrintInfoMobil().

Praktik 6a

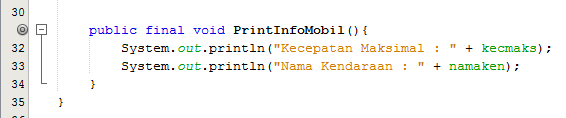
Penggunaan klausa final pada kelas Mobil



Program menjadi error karena suatu pewarisan(inheritance) tidak dapat dilakukan dari suatu kelas akhir(final).

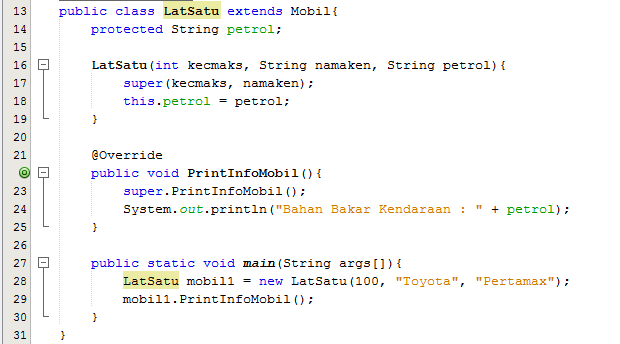
Praktik 6b

Penggunaan klausa final pada method



Program menjadi error karena suatu method akhir (final) tidak dapat lagi dilakukan override pada melalui kelas manapun karena sudah merupakan bentuk terakhir dari method tersebut. Contoh diatas adalah method PrintInfoMobil().

1. **PEMBAHASAN LATIHAN**



Dari latihan yang diminta mahasiswa diminta untuk membuat kelas baru yang merupakan kelas turunan dari kelas Mobil dan kemudian melakukan override pada salah satu method yang ada. Dari kelas diatas yang merupakan kelas turunan dari kelas Mobil dapat diperhatikan bahwa method yang di-*override* adalah method printInfoMobil() dengan menambahkan informasi Bahan Bakar Kendaraan kedalam program.

1. **TUGAS**
2. **KESIMPULAN**

Mahasiswa mampu membuat suatu kelas turunan (pewarisan) yang mana dapat memudahkan pembuatan atau pengelompokan objek yang ada pada program. Kelas yang menjadi induk(superclass) dapat diakses juga melalui kelas anak(subclass) dengan memanfaatkan keyword super serta jika ingin melakukan perubahan pada kelas induk maka perlu melakukan override pada *subclass*.

1. **LISTING**

*Terlampir*