

TUGAS 8

Nama: Mohammad Dwi Anshor Al Ghifari

Kelas: PTI - A

Nim: 235150607111008

Asisten: 1. Kak Devan

2. Kak Rivaldo

Data dan Analisis hasil percobaan Pertanyaan

1. Jelaskan apa fungsi dari *extends* dan *super* pada kode subclass?

Jawab: Pada java fungsi *extend* dan fungsi *super* memiliki sebuah peran yang sangat penting dalam melakukan sebuah definisi hubungan antar antar kelas, khususnya dalam sebuah konteks pewarisan. Yang dimaksud warisan adalah konsep yang dimiliki pada sistem pemrograman berorientasi objek, yang dapat memungkinkan kelas baru tersebut mewarisi properti dan metode dari kelas yang telah ada.

- a. Fungsi *extends*: dapat membangun sebuah hubungan pewarisan antar subclass dan superclassnya sehingga subclass tersebut dapat mewarisi dan memanfaatkan anggota superclassnya.
- b. Fungsi *super*: menjadi sebuah penyedia sarana bagi subclass untuk mengakses dan berinteraksi dengan anggota dan konstruktor superclassnya

2. Untuk apa digunakan keyword *this* pada constructor, setter dan getter?

Jawab:

- a. *This* pada constructor: Kata kunci ini merujuk pada sebuah objek yang akan sedang dibangun. Sehingga memungkinkan untuk melakukan sebuah inisialisasi terhadap bidang objek dengan nilai yang sangat sesuai.
- b. *This* pada setter: Kata kunci ini guna untuk membedakan sebuah parameter metode (nilai yang baru) dan sebuah bidang instance dengan nama yang sama, dan juga dapat memperbarui suatu nama pada bidang instance objek dengan nilai yang diberikan melalui parameter tersebut.
- c. *This* pada getter: Kata kunci ini dapat digunakan secara eksplisit dalam getter, dikarenakan nilai ini dikembalikan secara langsung mewakili pada nilai bidang instance, dan juga dapat mengembalikan nilai bidang instance tanpa menggunakan *this*.

3. Tambahkan dan jalankan kode ini di kelas Main, lalu amati apa yang terjadi? `HeroAgility hero3 = (HeroAgility) heroUp; hero3.display();`

Jawab:

The screenshot shows an IDE with a Java project named 'Praktikum'. The 'Main.java' file is open, showing a class with a main method. The code includes comments and a line that attempts to cast a `HeroIntel` object to a `HeroAgility` object, which results in a `ClassCastException`. The error message in the console reads: 'Exception in thread "main" java.lang.ClassCastException: class HeroIntel cannot be cast to class HeroAgility (HeroIntel and HeroAgility are in unnamed module of loader 'app')'. The console also shows the output of the program: 'Ucup is a Intel Hero.', 'Ucup is a Intel Hero.', 'Boy is a regular hero.', and 'Ucup is a Intel Hero.'.

Jika ditambahkan kode program tersebut maka akan menyebabkan `ClassCastException` yang dikarenakan exception sudah terjadi sebelumnya.

- Ubahlah modifier atribut type pada class `HeroIntel` dan `HeroAgility` menjadi `public`, lalu coba akses langsung melalui class `Main`. Apakah atribut bisa diakses langsung atau tidak, jelaskan!

Jawab:

- Class `HeroIntel`

The screenshot shows the `HeroIntel.java` file in the IDE. The code defines a `HeroIntel` class that extends a `Hero` class. It has a `type` attribute of type `String` and a `display` method that prints the hero's name and type. The code is as follows:

```
Tugas 8 > HeroIntel.java > HeroIntel > type
Click here to ask Blackbox to help you code faster
1 public class HeroIntel extends Hero {
2     public String type;
3
4     public HeroIntel(String name, double health) {
5         super(name, health);
6         this.type = "Intel";
7     }
8
9     public void display() {
10        System.out.println(this.getName() + " is a " + this.type + " Hero.");
11    }
12 }
```

- Class `HeroAgility`

```

Tugas 8 > J HeroAgility.java > HeroAgility
Click here to ask Blackbox to help you code faster
1 public class HeroAgility extends Hero {
2     public String type;
3
4     public HeroAgility(String name, double health) {
5         super(name, health);
6         this.type = "Agility";
7     }
8
9     public void display() {
10        System.out.println(this.getName() + " is a " + this.type + " Hero.");
11    }
12 }

```

c. Main

```

Tugas 8 > J Main.java > ...
Click here to ask Blackbox to help you code faster
1 public class Main {
2     Run | Debug
3     public static void main(String[] args) {
4
5         HeroIntel hero1 = new HeroIntel(name:"Ucup", health:100);
6         hero1.display();
7
8         // Akses langsung ke type
9         System.out.println("Hero1 type: " + hero1.type);
10
11        HeroAgility hero2 = new HeroAgility(name:"Budi", health:200);
12        hero2.display();
13
14        // Akses langsung ke type
15        System.out.println("Hero2 type: " + hero2.type);
16    }
17 }

```

Atribut dapat diakses secara langsung dari kelas lain, termasuk dari 'Main.java'. Sehingga dapat dimungkinkan untuk dapat melihat dan dapat mengubah nilai 'type' dari luar kelas 'HeroIntel' dan 'HeroAgility', yang sebelumnya tidak dapat diakses dikarenakan modifier tersebut bertipe private. Jika memodifikasi sebuah atribut tersebut secara langsung dapat berguna, dan juga harus perlu diwaspadai dikarenakan dapat mengurangi sebuah keamanan dari kesederhanaan kode tersebut dengan mengizinkan akses langsung ke dalam sebuah data yang internal. Dalam hal ini dapat melanggar sebuah prinsip dari enkapsulasi dalam sebuah pemrograman berorientasi objek.

5. Buatlah class baru HeroMagic dengan atribut tambahan power = "Magic" serta extends semua atribut dan method dari class Hero. Kemudian coba buatlah kode untuk upcasting dan downcasting dari class HeroMagic ke Hero pada class Main!

Jawab:

- a. HeroMagic

```
Tugas 8 > HeroMagic.java > ...
  Click here to ask Blackbox to help you code faster
1 public class HeroMagic extends Hero {
2     private String power;
3
4     public HeroMagic(String name, double health) {
5         super(name, health);
6         this.power = "Magic";
7     }
8
9     @Override
10    public void display() {
11        System.out.println(this.getName() + " is a " + this.power + " Hero.");
12    }
13 }
14
```

- b. Main

```
Tugas 8 > J Main.java > ...
Click here to ask Blackbox to help you code faster
1 public class Main {
  Run | Debug
2   public static void main(String[] args) {
3
4       HeroIntel hero1 = new HeroIntel(name:"Ucup", health:100);
5       hero1.display();
6
7       // Upcasting dari HeroIntel ke Hero
8       Hero heroUp = (Hero) hero1;
9       heroUp.display();
10
11      Hero heroReg = new Hero(name:"Boy", health:100);
12      heroReg.display();
13
14      // Downcasting kembali dari Hero ke HeroIntel
15      HeroIntel hero2 = (HeroIntel) heroUp;
16      hero2.display(); // Ini berhasil downcasting
17
18      // Membuat objek HeroMagic
19      HeroMagic magicHero = new HeroMagic(name:"Zara", health:120);
20      magicHero.display();
21
22      // Upcasting dari HeroMagic ke Hero
23      Hero heroMagicUp = (Hero) magicHero;
24      heroMagicUp.display();
25
26      // Downcasting kembali dari Hero ke HeroMagic
27      HeroMagic magicHeroDown = (HeroMagic) heroMagicUp;
28      magicHeroDown.display(); // Ini berhasil downcasting
29  }
30 }
31
```

c. Output

```
PROBLEMS 40 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS C:\Users\FLASH\OneDrive\Dokumen\PBO\Praktikum> & 'C:\Program Files\Java\
rs\FLASH\AppData\Roaming\Code\User\workspaceStorage\e293a9c818672859c4683d
Ucup is a Intel Hero.
Ucup is a Intel Hero.
Boy is a regular hero.
Ucup is a Intel Hero.
Zara is a Magic Hero.
Zara is a Magic Hero.
Zara is a Magic Hero.
PS C:\Users\FLASH\OneDrive\Dokumen\PBO\Praktikum>
```