

Zid'Avwa Al Bari'i

244107020083 / TI – 2I / 26

All the code here is on GitHub

<https://github.com/ZidAvwa/CollegeStudy/tree/main/3rdSemester/11.%20Jobsheet11>

Jobsheet 11

Exercise 1

Codes

```
src > J ICumlaude.java > ICumlaude
1 public interface ICumlaude {
2     public abstract void lulus();
3     public abstract void meraihIPKTinggi();
4 }

src > J Mahasiswa.java > Mahasiswa
1 public class Mahasiswa {
2     protected String nama;
3
4     public Mahasiswa(String nama) { this.nama = nama; }
5     public void kuliahDiKampus() {
6         System.out.println("Aku mahasiswa, namanya " + this.nama);
7         System.out.println(x: "Aku berkuliah di kampus");
8     }
9 }

src > J Sarjana.java > Sarjana
1 public class Sarjana extends Mahasiswa implements ICumlaude {
2     public Sarjana(String nama) { super(nama); }
3
4     @Override
5     public void lulus() { System.out.println("Sua sudah menyelesaikan SKRIPSI"); }
6     @Override
7     public void meraihIPKTinggi() { System.out.println("IPK-ku lebih dari 3,51"); }
8 }

src > J Pascasarjana.java > Pascasarjana
1 public class Pascasarjana extends Mahasiswa implements ICumlaude {
2     public Pascasarjana(String nama) { super(nama); }
3
4     @Override
5     public void lulus() { System.out.println("Sua sudah menyelesaikan TESIS"); }
6     @Override
7     public void meraihIPKTinggi() { System.out.println("IPK-ku lebih dari 3,71"); }
8 }

src > J Rektor.java > Rektor
1 public class Rektor {
2     public void beriSertifikatCumlaude(ICumlaude mahasiswa) {
3         System.out.println("Saya REKTOR, memberikan sertifikat cumlaude.");
4         System.out.println("Selamat! silahkan memperkenalkan diri anda.");
5
6         mahasiswa.lulus();
7         mahasiswa.rahaiIPKTinggi();
8     }
9 }

src > J Interfacemain.java > Interfacemain
1 public class Interfacemain {
2     Run/Debug
3     public static void main(String[] args) {
4         Rektor pakrektor = new Rektor();
5
6         Mahasiswa mhsBiasa = new Mahasiswa(nama: "Charlie");
7         Sarjana sarjanaCumlaude = new Sarjana(nama: "Dini");
8         Pascasarjana masterCumlaude = new Pascasarjana(nama: "Elok");
9
10        pakrektor.beriSertifikatCumlaude(mhsBiasa);
11        pakrektor.beriSertifikatCumlaude(sarjanaCumlaude);
12        pakrektor.beriSertifikatCumlaude(masterCumlaude);
13    }
14 }

The method beriSertifikatCumlaude(ICumlaude) in the type Rektor is not applicable for the arguments (Mahasiswa) Java(67108979) [Ln 9, Col 19]
```

To fix the error just remove mhsBiasa

```
src > J Interfacemain.java > Interfacemain
1 public class Interfacemain {
2     Run/Debug
3     public static void main(String[] args) {
4         Rektor pakrektor = new Rektor();
5
6         // Mahasiswa mhsBiasa = new Mahasiswa("Charlie");
7         Sarjana sarjanaCumlaude = new Sarjana(nama: "Dini");
8         Pascasarjana masterCumlaude = new Pascasarjana(nama: "Elok");
9
10        pakrektor.beriSertifikatCumlaude(sarjanaCumlaude);
11        pakrektor.beriSertifikatCumlaude(masterCumlaude);
12    }
13 }

Saya REKTOR, memberikan sertifikat cumlaude.
Selamat! silahkan perkenalkan diri anda.
Aku sudah menyelesaikan SKRIPSI
IPK-ku lebih dari 3,51
-----
Saya REKTOR, memberikan sertifikat cumlaude.
Selamat! silahkan perkenalkan diri anda.
Aku sudah menyelesaikan TESIS
IPK-ku lebih dari 3,71
-----
```

PRACTICE QUESTION 1

1. In step 9, why is there a warning on the 3rd line?

The warning on import `interfacelatihan.ICumlaude;` appears because the `ICumlaude` interface was imported but never used directly in that specific file (the `Program` class). The IDE is just letting you know the import isn't necessary.

2. What happens if the 3rd line (import) is omitted?

If you remove the import, and any code in that file tries to explicitly use the `ICumlaude` type (for example, `ICumlaude mhs;`), the program will fail to compile. It won't know what `ICumlaude` is.

3. Why did an error occur in step number 9?

The beriSertifikatCumlaude method requires a parameter that implements the ICumlaude interface. The mhsBiasa object is just a plain Mahasiswa. If you look at the Mahasiswa class definition, it does not implement ICumlaude.

4. Can the kuliahDiKampus() method be called from a Sarjana object?

Yes. The Sarjana class extends the Mahasiswa class. Because of inheritance, Sarjana automatically gets all the public methods from Mahasiswa, including kuliahDiKampus().

5. Can kuliahDiKampus() be called from the mahasiswa parameter inside the Rektor's beriSertifikatCumlaude method?

No. The parameter is declared as type ICumlaude. The compiler only allows you to call methods defined in the ICumlaude interface (lulus() and meraihIPKTinggi()), even if the object passed in also has other methods.

6. Modify the beriSertifikatCumlaude method in the Rektor class for the new output.

```
1 public class Rektor {
2     public void beriSertifikatCumlaude(ICumlaude mahasiswa) {
3         System.out.println("Saya REKTOR, memberikan sertifikat cumlaude.");
4         System.out.println("Selamat! silahkan berkenalan diri anda ...");
5
6         ((Mahasiswa)mahasiswa).kuliahDiKampus();
7         mahasiswa.lulus();
8         mahasiswa.meraihIPKTinggi();
9
10        System.out.println("-----");
11    }
12 }
```

```
Saya REKTOR, memberikan sertifikat cumlaude.
Selamat! silahkan berkenalan diri anda ...
Aku mahasiswa, namaku Dini
Aku berkuliah di kampus
Aku sudah menyelesaikan SKRIPSI
IPK-ku lebih dari 3,51
-----
Saya REKTOR, memberikan sertifikat cumlaude.
Selamat! silahkan berkenalan diri anda ...
Aku mahasiswa, namaku Elok
Aku berkuliah di kampus
Aku sudah menyelesaikan TESIS
IPK-ku lebih dari 3,71
-----
```

Exercise 2

Modified Code

```
1 public interface IBerprestasi {
2     public abstract void menjuaraiKompetisi();
3     public abstract void membuatPublikasiIlmiah();
4 }

1 public class Pascasarjana extends Mahasiswa implements ICumlaude, IBerprestasi {
2     public Pascasarjana(String nama) { super(nama); }
3
4     @Override
5     public void lulus() { System.out.println("Aku sudah menyelesaikan TESIS"); }
6
7     @Override
8     public void meraihIPKTinggi() { System.out.println("IPK-ku lebih dari 3,71"); }
9
10    @Override
11    public void menjuaraiKompetisi() { System.out.println("Saya telah menjuarai kompetisi INTERNASIONAL"); }
12
13    @Override
14    public void membuatPublikasiIlmiah() { System.out.println("Saya menerbitkan artikel di jurnal INTERNASIONAL"); }
15 }
```

```
1 public class Interfacemhs {
2     public static void main(String[] args) {
3         Rektor pakRektor = new Rektor();
4
5         // Mahasiswa mhsBiasa = new Mahasiswa("Charlie");
6         Sarjana sarjanaCumlaude = new Sarjana(nama: "Dini");
7         Pascasarjana masterCumlaude = new Pascasarjana(nama: "Elok");
8
9         // pakRektor.beriSertifikatCumlaude(mhsBiasa);
10        // pakRektor.beriSertifikatCumlaude(sarjanaCumlaude);
11        // pakRektor.beriSertifikatCumlaude(masterCumlaude);
12        pakRektor.beriSertifikatMagang(sarjanaCumlaude); // The new
13        pakRektor.beriSertifikatMagang(masterCumlaude);
14    }
15 }
```

To fix the error just do the same for the Sarjana.java

```
src > Sarjana.java > Sarjana
1 public class Sarjana extends Mahasiswa implements ICumlaude, IBerprestasi {
2     public Sarjana(String nama) { super(nama); }
3
4     @Override
5     public void lulus() { System.out.println(x: "Aku sudah menyelesaikan SKRIPSI"); }
6     @Override
7     public void meraihIPKtinggi() { System.out.println(x: "IPK-ku lebih dari 3,51"); }
8
9     @Override
10    public void menjuaraiKompetisi() { System.out.println(x: "Saya telah menjuarai kompetisi NASIONAL"); }
11    @Override
12    public void membuatPublikasiIlmiah() { System.out.println(x: "Saya menerbitkan artikel di Jurnal NASIONAL"); }
13 }
```

```
Saya REKTOR, memberikan sertifikat MAWAPRES.
Selamat! Bagaimana Anda bisa berprestasi?
Aku mahasiswa, namaku Dini
Aku berkuliah di kampus
Saya telah menjuarai kompetisi NASIONAL
Saya menerbitkan artikel di Jurnal NASIONAL
-----
Saya REKTOR, memberikan sertifikat MAWAPRES.
Selamat! Bagaimana Anda bisa berprestasi?
Aku mahasiswa, namaku Elok
Aku berkuliah di kampus
Saya telah menjuarai kompetisi INTERNASIONAL
Saya menerbitkan artikel di Jurnal INTERNASIONAL
-----
```

PRACTICE QUESTION 2

1. **From the script changes, what happened and explain the reason?**
Any class that implements IBerprestasi is now forced to provide its own code for the menjuaraiKompetisi() and membuatPublikasiIlmiah() methods.
2. **Explain why the script code IBerprestasi prestasi = new IBerprestasi(); is wrong.**
cannot create an instance (an object) directly from an interface. An interface is just an abstract template or a set of rules. You can only create objects from a concrete class (like Sarjana or PascaSarjana) that implements the interface.
3. **Match the output**

To fix the error just do the same for the Sarjana.java

```
src > Sarjana.java > Sarjana
1 public class Sarjana extends Mahasiswa implements ICumlaude, IBerprestasi {
2     public Sarjana(String nama) { super(nama); }
3
4     @Override
5     public void lulus() { System.out.println(x: "Aku sudah menyelesaikan SKRIPSI"); }
6     @Override
7     public void meraihIPKtinggi() { System.out.println(x: "IPK-ku lebih dari 3,51"); }
8
9     @Override
10    public void menjuaraiKompetisi() { System.out.println(x: "Saya telah menjuarai kompetisi NASIONAL"); }
11    @Override
12    public void membuatPublikasiIlmiah() { System.out.println(x: "Saya menerbitkan artikel di Jurnal NASIONAL"); }
13 }
```

```
Saya REKTOR, memberikan sertifikat MAWAPRES.
Selamat! Bagaimana Anda bisa berprestasi?
Aku mahasiswa, namaku Dini
Aku berkuliah di kampus
Saya telah menjuarai kompetisi NASIONAL
Saya menerbitkan artikel di Jurnal NASIONAL
-----
Saya REKTOR, memberikan sertifikat MAWAPRES.
Selamat! Bagaimana Anda bisa berprestasi?
Aku mahasiswa, namaku Elok
Aku berkuliah di kampus
Saya telah menjuarai kompetisi INTERNASIONAL
Saya menerbitkan artikel di Jurnal INTERNASIONAL
-----
```

Assignment

Code


```
src > J IHerbivora.java > IHerbivora
1 public interface IHerbivora {
2     public void displayMakan();
3 }
```

```
src > J IKarnivora.java > IKarnivora
1 public interface IKarnivora {
2     public void displayMakan();
3 }
```

```
src > J Binatang.java > Binatang
1 public abstract class Binatang {
2     protected String nama;
3     protected int jmlKaki;
4
5     public Binatang(String nama, int jmlKaki) {
6         this.nama = nama;
7         this.jmlKaki = jmlKaki;
8     }
9
10    public void setNama(String nama) { this.nama = nama; }
11    public String getNama() { return nama; }
12    public int getKaki() { return jmlKaki; }
13
14    public abstract void displayBinatang();
15 }
```

```
src > J Singa.java > Singa
1 public class Singa extends Binatang implements IKarnivora {
2     private String suara;
3     private String warnaBulu;
4
5     public Singa(String nama, int jmlKaki, String suara, String warnaBulu) {
6         super(nama, jmlKaki);
7         this.suara = suara;
8         this.warnaBulu = warnaBulu;
9     }
10
11    @Override
12    public void displayMakan() {
13        System.out.println("Jenis: Karnivora");
14        System.out.println("Makanan: Daging");
15    }
16
17    @Override
18    public void displayBinatang() {
19        System.out.println("Nama: " + this.nama);
20        System.out.println("Jumlah Kaki: " + this.jmlKaki);
21    }
22
23    public void displayData() {
24        displayMakan();
25        displayBinatang();
26        System.out.println("Suara: " + this.suara);
27        System.out.println("Warna Bulu: " + this.warnaBulu);
28    }
29 }
```

```
src > J Kadal.java > Kadal
1 public class Kadal extends Binatang implements IHerbivora {
2     private String suara;
3     private String warnaBulu;
4
5     public Kadal(String nama, int jmlKaki, String suara, String warnaBulu) {
6         super(nama, jmlKaki);
7         this.suara = suara;
8         this.warnaBulu = warnaBulu;
9     }
10
11    @Override
12    public void displayMakan() {
13        System.out.println("Jenis: Herbivora");
14        System.out.println("Makanan: Tumbuhan");
15    }
16
17    @Override
18    public void displayBinatang() {
19        System.out.println("Nama: " + this.nama);
20        System.out.println("Jumlah Kaki: " + this.jmlKaki);
21    }
22
23    public void displayData() {
24        displayMakan();
25        displayBinatang();
26        System.out.println("Suara: " + this.suara);
27        System.out.println("Warna Bulu: " + this.warnaBulu);
28    }
29 }
```

```
src > J Gajah.java > Gajah
1 public class Gajah extends Binatang implements IKarnivora, IHerbivora {
2     private String suara;
3     private String warnaBulu;
4
5     public Gajah(String nama, int jmlKaki, String suara, String warnaBulu) {
6         super(nama, jmlKaki);
7         this.suara = suara;
8         this.warnaBulu = warnaBulu;
9     }
10
11    @Override
12    public void displayMakan() {
13        System.out.println("Jenis: Karnivora + Herbivora");
14        System.out.println("Makanan: Daging + Tumbuhan");
15    }
16
17    @Override
18    public void displayBinatang() {
19        System.out.println("Nama: " + this.nama);
20        System.out.println("Jumlah Kaki: " + this.jmlKaki);
21    }
22
23    public void displayData() {
24        displayMakan();
25        displayBinatang();
26        System.out.println("Suara: " + this.suara);
27        System.out.println("Warna Bulu: " + this.warnaBulu);
28    }
29 }
```

Jenis: Herbivora
Makanan: Tumbuhan
Nama: Kadal
Jumlah Kaki: 4
Suara: Hehehehe
Warna Bulu: Abu-Abu

Jenis: Karnivora + Herbivora
Makanan: Daging + Tumbuhan
Nama: Gajah
Jumlah Kaki: 4
Suara: Haaum Haaum
Warna Bulu: Hitam Manis

Jenis: Karnivora
Makanan: Daging
Nama: Singa
Jumlah Kaki: 4
Suara: Roaar Roaar
Warna Bulu: Coklat

```
src > J Menu.java > Menu
1 public class Menu {
2     public static void main(String[] args) {
3         System.out.println("Menu");
4
5         Kadal k = new Kadal("Kadal", 4, suara: "Hehehehe", warnaBulu: "Abu-Abu");
6         k.displayData();
7         System.out.println();
8
9         Gajah g = new Gajah("Gajah", 4, suara: "Haaum Haaum", warnaBulu: "Hitam Manis");
10        g.displayData();
11        System.out.println();
12
13        Singa s = new Singa("Singa", 4, suara: "Roaar Roaar", warnaBulu: "Coklat");
14        s.displayData();
15    }
16 }
```

UML

