

LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS OBJEK
JOBSHEET 11
(Interface dalam OOP Java)



Oleh :

Debora Febriana P. A
TI2B / 08
1941720156

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN 2020/2021

Percobaan 1

1. Class ICumlaude

```
6 package Interface;
7
8 /**
9  *
10  * @author DEBORA
11  */
12 public interface ICumlaude {
13     public abstract void lulus();
14     public abstract void meraihIPKTinggi();
15 }
16
17
```

2. Class Mahasiswa

```
10  * @author DEBORA
11  */
12 public class Mahasiswa {
13     protected String nama;
14     public Mahasiswa(String nama){
15         this.nama=nama;
16     }
17
18     public void kuliahDiKampus(){
19         System.out.println("Aku mahasiswa, namaku "+this.nama);
20         System.out.println("Aku berkuliah di kampus.");
21     }
22 }

```

3. Class Sarjana

```
10  * @author DEBORA
11  */
12 public class Sarjana extends Mahasiswa implements ICumlaude{
13     public Sarjana(String nama){
14         super(nama);
15     }
16     @Override
17     public void lulus(){
18         System.out.println("Aku sudah menyelesaikan SKRIPSI");
19     }
20
21     @Override
22     public void meraihIPKTinggi() {
23         System.out.println("IPK-ku lebih dari 3,51");
24     }
25 }

```

4. Class PascaSarjana

```
10 | * @author DEBORA
11 | */
12 | public class PascaSarjana extends Mahasiswa implements ICumlaude{
13 |     public PascaSarjana(String nama){
14 |         super(nama);
15 |     }
16 |
17 |     @Override
18 |     public void lulus(){
19 |         System.out.println("Aku sudah menyelesaikan TESIS");
20 |     }
21 |     @Override
22 |     public void meraihIPKTinggi(){
23 |         System.out.println("IPK-ku lebih dari 3,71");
24 |     }
25 |
26 | }
```

5. Class Rektor

```
10 | * @author DEBORA
11 | */
12 | public class Rektor {
13 |     public void beriSertifikatCumlaude(ICumlaude mahasiswa){
14 |         System.out.println("Saya REKTOR, memberikan sertifikat cumlaude");
15 |         System.out.println("Selamat! Silahkan perkenalkan diri Anda...");
16 |
17 |         mahasiswa.lulus();
18 |         mahasiswa.meraihIPKTinggi();
19 |
20 |         System.out.println("-----");
21 |     }
22 | }
```

6. Class interfacemain

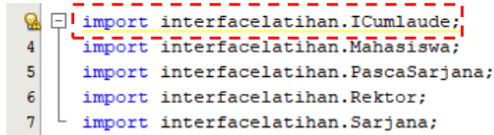
```
10      * @author DEBORA
11      */
12      import Interface.ICumlaude;
13      import Interface.Mahasiswa;
14      import Interface.PascaSarjana;
15      import Interface.Rektor;
16      import Interface.Sarjana;
17
18      public class interfacemain{
19          public static void main(String[] args){
20              Rektor pakrektor = new Rektor();
21              Mahasiswa mhsBiasa = new Mahasiswa ("Charlie");
22              Sarjana sarjanaCumlaude = new Sarjana("Dini");
23              PascaSarjana masterCumlaude = new PascaSarjana("Elok");
24
25              //      pakrektor.beriSertifikatCumlaude(mhsBiasa);
26              pakrektor.beriSertifikatCumlaude(sarjanaCumlaude);
27              pakrektor.beriSertifikatCumlaude(masterCumlaude);
28          }
29      }
```

7. Output

```
compile-single:
run-single:
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
-----
BUILD SUCCESSFUL (total time: 1 second)
```

PERTANYAAN PERCOBAAN 1

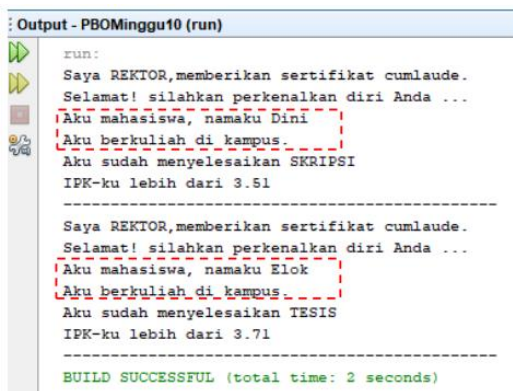
- 1) Pada langkah ke 9, pada baris program ke 3 terdapat warning pada script tersebut.
Jelaskan penyebab terjadinya hal tersebut ?



```
1 import interfacelatihan.ICumlaude;  
4 import interfacelatihan.Mahasiswa;  
5 import interfacelatihan.PascaSarjana;  
6 import interfacelatihan.Rektor;  
7 import interfacelatihan.Sarjana;
```

The screenshot shows a Java IDE with a warning icon on line 3. The code imports several interfaces and classes from the package 'interfacelatihan'. The warning is likely due to the 'import interfacelatihan.ICumlaude;' statement, as it is not a standard Java interface.

- 2) Pada langkah ke 9, pada baris program ke 3. Apa yang terjadi jika script tersebut dihilangkan? Jelaskan menurut pemahaman anda.
- 3) Mengapa pada langkah nomor 9 terjadi error? Jelaskan!
- 4) Dapatkah method **kuliahDiKampus()** dipanggil dari objek **sarjanaCumlaude** di class **Program**? Mengapa demikian?
- 5) Dapatkah method **kuliahDiKampus()** dipanggil dari parameter **mahasiswa** di method **beriSertifikatCumlaude()** pada class **Rektor**? Mengapa demikian?
- 6) Modifikasilah method **beriSertifikatCumlaude()** pada class **Rektor** agar hasil eksekusi class **Program** menjadi seperti berikut ini:



```
: Output - PBOMinggu10 (run)  
  
run:  
Saya REKTOR,memberikan sertifikat cumlaude.  
Selamat! silahkan perkenalkan 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 perkenalkan diri Anda ...  
Aku mahasiswa, namaku Elok  
Aku berkuliah di kampus  
Aku sudah menyelesaikan TESIS  
IPK-ku lebih dari 3.71  
-----  
BUILD SUCCESSFUL (total time: 2 seconds)
```

The screenshot shows the output of a Java program. It displays two successful runs. In the first run, a student named Dini is introduced, and the program outputs their name, university, and GPA. In the second run, a student named Elok is introduced, and the program outputs their name, university, and GPA. The output is formatted with dashed lines separating the two runs.

Jawaban

1. Warning pada script muncul dikarenakan import yang dilakukan tidak terpakai.
2. Saat script dihilangkan program tidak mengalami error saat dijalankan.
3. Pada langkah 9 terjadi error dikarenakan tidak ada implement interface pada class Rektor.
4. Method **kuliahDiKampus()** dapat dipanggil dari objek **sarjanaCumlaude**. Hal ini dikarenakan class **Sarjana** meng-extends class **Mahasiswa** yang menjadikannya memiliki turunan dari class **parentnya**.
5. Method **kuliahDiKampus()** tidak dapat dipanggil dari parameter **mahasiswa** di method **beriSertifikatCumlaude()** pada class **Rektor**. Class **Rektor** tidak dapat mengakses method **kuliahDiKampus()** dari class **Mahasiswa**. Program mengalami error.

6. Modifikasi program

- Melakukan modifikasi pada Class ICumlaude

```
public interface ICumlaude {  
    public abstract void lulus();  
    public abstract void meraihIPKTinggi();  
    public void kuliahDiKampus();  
}
```

- Melakukan modifikasi pada Class Rektor. Memanggil method kuliahDiKampus();

```
public class Rektor {  
    public void beriSertifikatCumlaude(ICumlaude mahasiswa){  
        System.out.println("Saya REKTOR, memberikan sertifikat cumlaude");  
        System.out.println("Selamat! Silahkan perkenalkan diri Anda...");  
  
        mahasiswa.kuliahDiKampus();  
        mahasiswa.lulus();  
        mahasiswa.meraihIPKTinggi();  
  
        System.out.println("-----");  
    }  
}
```

- Hasil

```
compile-single:  
run-single:  
Saya REKTOR, memberikan sertifikat cumlaude  
Selamat! Silahkan perkenalkan 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 perkenalkan diri Anda...  
Aku mahasiswa, namaku Elok  
Aku berkuliah di kampus.  
Aku sudah menyelesaikan TESIS  
IPK-ku lebih dari 3,71  
-----  
BUILD SUCCESSFUL (total time: 1 second)
```

Percobaan 2

1. Class Iberprestasi

```
10  | * @author DEBORA
11  | */
    |
    | public interface IBerprestasi {
    |     public abstract void menjuaraiKompetisi();
    |     public abstract void membuatPublikasiIlmiah();
15  | }
```

2. Class PascaSarjana

```
10  | * @author DEBORA
11  | */
12  | public class PascaSarjana extends Mahasiswa implements ICumlaude, IBerprestasi {
13  |     public PascaSarjana(String nama){
14  |         super(nama);
15  |     }
16  |
17  |     @Override
18  |     public void menjuaraiKompetisi(){
19  |         System.out.println("Saya telah menjuarai kompetisi INTERNASIONAL");
20  |     }
21  |     @Override
22  |     public void membuatPublikasiIlmiah(){
23  |         System.out.println("Saya menerbitkan artikel di jurnal INTERNASIONAL");
24  |     }
```

3. Class Rektor

```
10  | * @author DEBORA
11  | */
12  | public class Rektor {
13  |     // public void beriSertifikatCumlaude(ICumlaude mahasiswa){
14  |     public void beriSertifikatMawapres(IBerprestasi mahasiswa){
15  |         System.out.println("Saya REKTOR, memberikan sertifikat MAWAPRES");
16  |         System.out.println("Selamat! Bagaimana Anda bisa berprestasi?");
17  |
18  |         mahasiswa.menjuaraiKompetisi();
19  |         mahasiswa.membuatPublikasiIlmiah();
20  |
21  |         System.out.println("-----");
```

4. Class MultipleInterface

```
10 |      * @author DEBORA
11 |      */
12 |      public class MultipleInterface {
13 |          public static void main(String [] args){
14 |              Rektor pakrektor = new Rektor();
15 |              Sarjana sarjanaCum = new Sarjana("Dini");
16 |              PascaSarjana masterCum = new PascaSarjana("Elok");
17 |
18 |              //      pakrektor.beriSertifikatMawapres(sarjanaCum);
19 |              pakrektor.beriSertifikatMawapres(masterCum);
20 |          }
21 |      }
```

5. Output

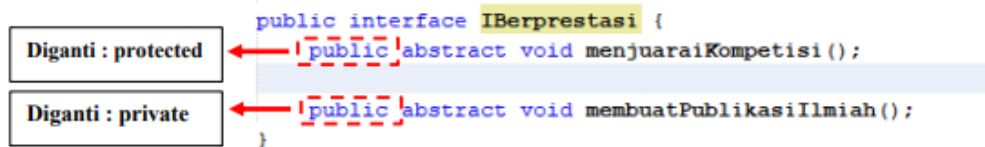
compile-single:

run-single:

```
Saya REKTOR, memberikan sertifikat MAWAPRES
Selamat! Bagaimana Anda bisa berprestasi?
Saya telah menjuarai kompetisi INTERNASIONAL
Saya menerbitkan artikel di jurnal INTERNASIONAL
-----
BUILD SUCCESSFUL (total time: 2 seconds)
```


Pertanyaan

1. Pada script code interface **IBerprestasi**, modifikasi script tersebut sesuai dengan gambar dibawah ini :



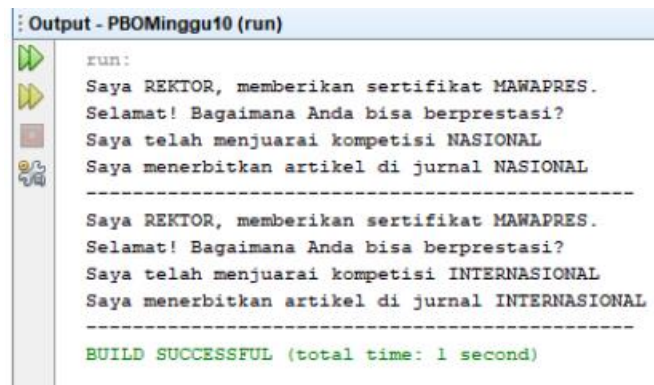
Dari perubahan script diatas, apa yang terjadi ? serta jelaskan alasannya (capture hasilnya)

2. Perhatikan script code dibawah ini :

```
22 public static void main(String[] args) {  
    IBerprestasi prestasi = new IBerprestasi();  
}
```

Jelaskan menurut anda, mengapa hasil dari script code tersebut error ?

3. Apabila **Sarjana Berprestasi** harus **menjuarai kompetisi NASIONAL** dan **menerbitkan artikel di jurnal NASIONAL**, maka modifikasilah class-class yang terkait pada aplikasi Anda agar di class **Program** objek **pakRektor** dapat memberikan sertifikat mawapres pada objek **sarjanaCumlaude**.



Jawaban

1. Terjadi error pada modifier protected. Muncul pesan error “modifier protected not allowed here” dikarenakan modifier protected tidak diizinkan pada interface. Dan untuk modifier private akan muncul pesan error “illegal combination of modifiers: abstract and private”. Intinya untuk interface diharuskan menggunakan akses modifier public.
2. Script kode tersebut error karena interface hanya bisa di instansiasi melalui class yang mengimplementnya saja.

3. Modifikasi class agar objek pakRektor dapat memberikan sertifikat mawapres pada objek sarjanaCumlaude :

- Memodifikasi Class Sarjana dengan meng implementasikan interface IBerprestasi

```
27 //MULTIPLE INTERFACE
28 public class Sarjana extends Mahasiswa implements IBerprestasi{
```

- Modifikasi pada Class Main

```
17
18     pakrektor.beriSertifikatMawapres(sarjanaCum);
19     pakrektor.beriSertifikatMawapres(masterCum);
20 }
21 }
```

- Hasil

```
compile-single:
run-single:
Saya REKTOR, memberikan sertifikat MAWAPRES
Selamat! Bagaimana Anda bisa berprestasi?
Saya telah menjuarai kompetisi NASIONAL
Saya menerbitkan artikel di jurnal NASIONAL
-----
Saya REKTOR, memberikan sertifikat MAWAPRES
Selamat! Bagaimana Anda bisa berprestasi?
Saya telah menjuarai kompetisi INTERNASIONAL
Saya menerbitkan artikel di jurnal INTERNASIONAL
-----
BUILD SUCCESSFUL (total time: 1 second)
```

Tugas

1. Class Ikarnivora

```
6 package Interface.Tugas;
7
8 /**
9  *
10  * @author DEBORA
11  */
12 public interface IKarnivora {
13     public void displayMakan();
14 }
```

2. Class Iherbivora

```
10  * @author DEBORA
11  */
12 public interface IHerbivora {
13     public void displayMakan();
14 }
```

3. Class Binatang

```
10  * @author DEBORA
11  */
12 public abstract class Binatang {
13     protected String nama;
14     protected int jmlKaki;
15
16     public Binatang (String nama, int jmlKaki){
17         this.nama = nama;
18         this.jmlKaki = jmlKaki;
19     }
20     public void setNama(String nama){
21         this.nama=nama;
22     }
23     public String getNama(){
24         return nama;
25     }
26     public void setKaki(int jmlKaki){
27         this.jmlKaki = jmlKaki;
28     }
29     public int getKaki(){
30         return jmlKaki;
31     }
32     public void displayBinatang(){
33         System.out.println("Nama: "+getNama());
34         System.out.println("Jumlah kaki: "+getKaki());
35     }
36 }
```

4. Class Singa

```
10 | * @author DEBORA
11 | */
12 | public class Singa extends Binatang implements IKarnivora{
13 |     private String Suara;
14 |     private String warnaBulu;
15 |     public Singa(String Suara, String warnaBulu, String nama, int jmlKaki) {
16 |         super(nama, jmlKaki);
17 |         this.Suara = Suara;
18 |         this.warnaBulu = warnaBulu;
19 |     }
20 |     @Override
21 |     public void displayMakan() {
22 |         System.out.println("Makanan: Daging");
23 |     }
24 |     @Override
25 |     public void displayBinatang() {
26 |         System.out.println("Singa");
27 |     }
28 |     public void displayData(){
29 |         System.out.println("Jenis: Karnivora");
30 |         this.displayMakan();
31 |         System.out.println("Nama: " + super.nama);
32 |         System.out.println("Jumlah Kaki: " + super.jmlKaki);
33 |         System.out.println("Suara: " + this.Suara);
34 |         System.out.println("Warna Bulu: " + this.warnaBulu);
35 |         System.out.println();
36 |     }
37 | }
```

5. Class Keledai

```
10 | * @author DEBORA
11 | */
12 | public class Keledai extends Binatang implements IHerbivora{
13 |     private String suara;
14 |     private String warnaBulu;
15 |     public Keledai(String suara, String warnaBulu, String nama, int jmlKaki) {
16 |         super(nama, jmlKaki);
17 |         this.suara = suara;
18 |         this.warnaBulu = warnaBulu;
19 |     }
20 |     @Override
21 |     public void displayMakan() {
22 |         System.out.println("Makanan: Tumbuhan");
23 |     }
24 |     @Override
25 |     public void displayBinatang() {
26 |         System.out.println("Keledai");
27 |     }
28 |     public void displayData(){
29 |         System.out.println("Jenis: Herbivora");
30 |         this.displayMakan();
31 |         System.out.println("Nama: " + super.nama);
32 |         System.out.println("Jumlah Kaki: " + super.jmlKaki);
33 |         System.out.println("Suara: " + this.suara);
34 |         System.out.println("Warna Bulu: " + this.warnaBulu);
35 |         System.out.println();
36 |     }
37 | }
```

6. Class Gorila

```
10 | * @author DEBORA
11 | */
12 | public class Gorila extends Binatang implements IHerbivora, IKarnivora {
13 |     private String suara;
14 |     private String warnaBulu;
15 |     public Gorila(String suara, String warnaBulu, String nama, int jmlKaki) {
16 |         super(nama, jmlKaki);
17 |         this.suara = suara;
18 |         this.warnaBulu = warnaBulu;
19 |     }
20 |     @Override
21 |     public void displayMakan() {
22 |         System.out.println("Makanan: Daging + Tumbuhan");
23 |     }
24 |     @Override
25 |     public void displayBinatang() {
26 |         System.out.println("Gorilla");
27 |     }
28 |     public void displayData() {
29 |         System.out.println("Jenis: Karnivora + Herbivora");
30 |         this.displayMakan();
31 |         System.out.println("Nama: " + super.nama);
32 |         System.out.println("Jumlah Kak: " + super.jmlKaki);
33 |         System.out.println("Suara: " + this.suara);
34 |         System.out.println("Warna Bulu: " + this.warnaBulu);
35 |         System.out.println();
36 |     }
}
```

7. Class TugasMain

```
10 | * @author DEBORA
11 | */
12 | public class TugasMain {
13 |     public static void main(String[] args) {
14 |         Keledai keledai = new Keledai("HHHHHHHHHH", "ABU-ABU", "KEDELAI", 4);
15 |         Gorila gorila = new Gorila("HAAUM HAUUM", "HITAM MUANIS", "GULALIIII", 2);
16 |         Singa singa = new Singa("ROOARRRRRRR", "COKLATTT", "CINGACINGCINGCING", 4);
17 |
18 |         keledai.displayData();
19 |         gorila.displayData();
20 |         singa.displayData();
21 |     }
22 | }
```

8. Output

```
compile-single:
run-single:
Jenis: Herbivora
Makanan: Tumbuhan
Nama: KEDELAI
Jumlah Kaki: 4
Suara: HHHHHHHHHH
Warna Bulu: ABU-ABU

Jenis: Karnivora + Herbivora
Makanan: Daging + Tumbuhan
Nama: GULALIIII
Jumlah Kak: 2
Suara: HAAUM HAUUM
Warna Bulu: HITAM MUANIS

Jenis: Karnivora
Makanan: Daging
Nama: CINGACINGCINGCING
Jumlah Kaki: 4
Suara: ROOARRRRRRR
Warna Bulu: COKLATTT

BUILD SUCCESSFUL (total time: 1 second)
```