

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

JOBSHEET 11: INTERFACE CLASS



oleh :
Halim Teguh Saputro
2E
2141762122

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI**

POLITEKNIK NEGERI MALANG
Jl. Soekarno Hatta No .9, Jatimulyo, Kec. Lowokwaru, Kota Malang,
JawaTimur 65141

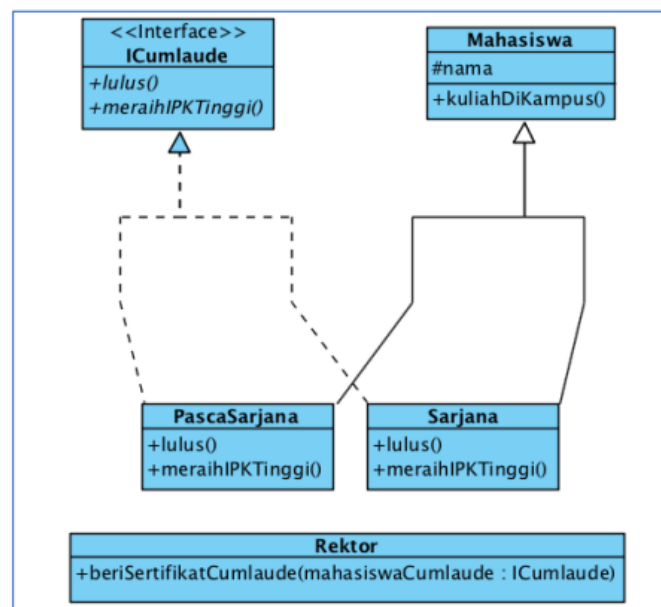
KOMPETENSI

Setelah melakukan percobaan pada modul ini, mahasiswa memahami konsep:

1. Menjelaskan maksud dan tujuan penggunaan Interface
2. Menerapkan Interface di dalam pembuatan program

PRAKTIKUM 1. INTERFACE

Pada sebuah wisuda, seorang Rektor akan memberikan penghargaan sertifikat *Cumlaude* pada semua mahasiswa yang memenuhi persyaratan. Persyaratan agar seorang mahasiswa dapat disebut sebagai *Cumlaude* berbeda-beda antara mahasiswa Sarjana dan Pasca Sarjana.



Untuk menjadi *cumlaude*, mahasiswa Sarjana harus mengerjakan **skripsi** dan memiliki IPK lebih tinggi dari 3,51. Sedangkan untuk mahasiswa Pasca Sarjana, mereka harus mengerjakan **tesis** dan meraih IPK lebih tinggi dari 3,71.

1. Pada project Latihan PBOMinggu10, buat sebuah package baru Bernama `interfacelatihan`. Kemudian klik Finish.
2. Pada package yang baru dibuat tersebut, tambahkan sebuah interface baru dengan cara klik kanan pada package > New > Java Interface... Beri nama interface baru tersebut dengan nama `ICumlaude`
3. Pada interface `ICumlaude` tersebut, tambahkan 2 abstract methods Bernama `lulus()` dan `meraihIPKTinggi()`.

```
src > interfaceLatihan > ICumlaude.java > ICumlaude
1 package interfaceLatihan;
2
3 public interface ICumlaude {
4     public abstract void lulus();
5
6     public abstract void meraihIPKTinggi();
7 }
```

4. Berikutnya, buatlah sebuah class baru Bernama Mahasiswa dengan baris-baris kode seperti dibawah ini.

```
src > interfaceLatihan > Mahasiswa.java > Mahasiswa > kuliahDiKampus()
1 package interfaceLatihan;
2
3 public class Mahasiswa {
4     protected String nama;
5
6     public Mahasiswa(String nama) {
7         this.nama = nama;
8     }
9
10    public void kuliahDiKampus() {
11        System.out.println("Aku mahasiswa, namaku " + this.nama);
12        System.out.println(x: "Aku berkuliah di kampus.");
13    }
14 }
```

5. Selanjutnya, buatlah class baru Bernama Sarjana yang merupakan turunan dari class Mahasiswa. Class Sarjana tersebut dibuat meng-Implements interface ICumlaude yang sudah dibuat sebelumnya tadi. Ketikkan kode di bawah pada class tersebut. Tips: anda dapat menggunakan fasilitas override otomatis dengan cara yang sama yaitu dengan mengklik ikon lampu peringatan seperti pada percobaan 1.

```
src > interfaceLatihan > Sarjana.java > Sarjana
1 package interfaceLatihan;
2
3 public class Sarjana extends Mahasiswa implements ICumlaude {
4     public Sarjana(String nama) {
5         super(nama);
6     }
7
8     @Override
9     public void lulus() {
10        // TODO Auto-generated method stub
11    }
```

```

13     @Override
14     public void meraihIPKtinggi() {
15         // TODO Auto-generated method stub
16     }
17
18 }

```

- Selanjutnya sesuaikan isi dari method lulus() dan meraihIPKtinggi() agar sama dengan baris kode di bawah.

```

src > interfaceLatihan > Sarjana.java > Sarjana > lulus()
1  package interfaceLatihan;
2
3  public class Sarjana extends Mahasiswa implements ICumlaude {
4      public Sarjana(String nama) {
5          super(nama);
6      }
7
8      @Override
9      public void lulus() {
10         // TODO Auto-generated method stub
11         System.out.println(x: "Aku sudah menyelesaikan SKRIPSI");
12     }
13
14     @Override
15     public void meraihIPKtinggi() {
16         // TODO Auto-generated method stub
17         System.out.println(x: "IPK-ku lebih dari 3,51");
18     }
19
20 }

```

Perhatikan pada baris kode di atas, class Sarjana meng-extend class Mahasiswa, ini berarti, Sarjana adalah Mahasiswa sementara itu agar semua objek dari class Sarjana ini nantinya dapat disebut sebagai Cumlaude makai a harus meng-Implements interface ICumlaude.

- Kemudian dengan cara yang sama buatlah class baru Bernama PascaSarjana dengan baris kode seperti di bawah ini.

```

src > interfaceLatihan > PascaSarjana.java > PascaSarjana
1  package interfaceLatihan;
2
3  public class PascaSarjana extends Mahasiswa implements ICumlaude {
4      public PascaSarjana(String nama) {
5          super(nama);
6      }

```

```

8      @Override
9      public void lulus() {
10         // TODO Auto-generated method stub
11         System.out.println(x: "Aku sudah menyelesaikan TESIS");
12     }
13
14     @Override
15     public void meraihIPKTinggi() {
16         // TODO Auto-generated method stub
17         System.out.println(x: "IPK-ku lebih dari 3,71");
18     }

```

8. Lalu buatlah sebuah class baru Bernama Rektor. Class ini adalah class yang memanfaatkan class-class Mahasiswa yang telah dibuat sebelumnya.

```

src > interfaceLatihan > J Rektor.java > R Rektor > R beriSertifikatCumlaude(ICumlaude)
1  package interfaceLatihan;
2
3  public class Rektor {
4      public void beriSertifikatCumlaude(ICumlaude mahasiswa) {
5          System.out.println(x: "Saya REKTOR, memberikan sertifikat cumlaude");
6          System.out.println(x: "Selamat! silahkan perkenalkan diri Anda..");
7
8          mahasiswa.lulus();
9          mahasiswa.meraihIPKTinggi();
10
11         System.out.println(x: "-----");
12     }
13 }

```

9. Terakhir, buatlah sebuah class baru Bernama interfaceMain yang diletakkan pada package yang sama dengan class-class percobaan 2. Tambahkan baris kode berikut ini:

```

src > J interfaceMain.java > R interfaceMain
1  import interfaceLatihan.Mahasiswa;
2  import interfaceLatihan.PascaSarjana;
3  import interfaceLatihan.Rektor;
4  import interfaceLatihan.Sarjana;
5
6  public class interfaceMain {
7      Run | Debug
8      public static void main(String[] args) {
9          Rektor pakRektor = new Rektor();
10
11         Mahasiswa mhsBiasa = new Mahasiswa(nama: "Halim");
12         Sarjana sarjanaCumlaude = new Sarjana(nama: "Teguh");
13         PascaSarjana masterCumlaude = new PascaSarjana(nama: "Saputro");

```

```

14     pakRektor.beriSertifikatCumlaude(mhsBiasa);
15     pakRektor.beriSertifikatCumlaude(sarjanaCumlaude);
16     pakRektor.beriSertifikatCumlaude(masterCumlaude);
17 }
18 }

```

10. Pada baris kode tersebut, apabila anda mengetikkan semua class dengan benar, maka akan terdapat error dan class Program tidak dapat dieksekusi. Perbaikilah kode anda agar program yang anda buat mengeluarkan output seperti berikut ini:

```

Output - PBOMinggu10 (run)

run:
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: 0 seconds)

```

```

src > J interfaceMain.java > interfaceMain > main(String[])
4  import interfaceLatihan.Sarjana;
5
6  public class interfaceMain {
7      Run | Debug
8      public static void main(String[] args) {
9          Rektor pakRektor = new Rektor();
10
11         Mahasiswa mhsBiasa = new Mahasiswa(nama: "Halim");
12         Sarjana sarjanaCumlaude = new Sarjana(nama: "Teguh");
13         PascaSarjana masterCumlaude = new PascaSarjana(nama: "Saputro");
14
15         // pakRektor.beriSertifikatCumlaude(mhsBiasa);
16         pakRektor.beriSertifikatCumlaude(sarjanaCumlaude);
17         pakRektor.beriSertifikatCumlaude(masterCumlaude);
18     }
19 }

```

Keterangan: saya mengubah baris ke-14 yaitu sintaks yang memanggil method `beriSertifikatCumlaude` yang memanggil `mhsBiasa` menjadi sintaks komentar.

```

er 3\5. Pemrograman Berbasis Objek\Praktikum\11
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
-----
PS C:\Users\Halim\Downloads\POLINEMA\Semester 3

```

PERTANYAAN PRAKTIKUM 1

1. Pada Langkah ke 9, pada baris program ke 3 terdapat warning pada script tersebut. Jelaskan penyebab terjadinya hal tersebut?
Jawab: terjadi warning karena program interfaceMain tidak menggunakan file ICumlaude. Sehingga seharusnya program tidak memerlukan import dari ICumlaude tersebut.
2. Pada Langkah ke 9, pada baris program ke 3. Apa yang terjadi jika script tersebut dihilangkan? Jelaskan pemahaman anda
Jawab: tidak terjadi masalah atau error, karena file ICumlaude tidak digunakan pada interfaceMain.
3. Mengapa pada langka nomor 9 terjadi error? Jelaskan!
Jawab: menurut saya, terjadi error karena method beriSertifikatCumlaude() memerlukan atau akan menjalankan method lulus() dan meraihIPKTinggi(). Namun pada class Mahasiswa tidak memiliki method tersebut.
4. Dapatkah method kuliahDiKampus() dipanggil dari objek sarjanaCumlaude di class Program? Mengapa demikian?
Jawab: Bisa, karena sarjanaCumlaude merupakan objek dari class Sarjana dan class Sarjana meng-extend ke class Mahasiswa yang memiliki method kuliahDiKampus().
5. Dapatkah method kuliahDiKampus() dipanggil dari parameter mahasiswa di method beriSertifikatCumlaude() pada class Rektor? mengapa demikian?
Jawab: Tidak bisa, karena parameter mahasiswa di method beriSertifikatCumlaude() itu berasal dari interface ICumlaude yang tidak memiliki method kuliahDiKampus(). Method kuliahDiKampus() berada pada class Mahasiswa bukan ICumlaude

6. Modifikasilah method berisertifikatCumlaude() pada class Rektor agar hasil eksekusi class Program menjadi seperti berikut ini:

a. Sarjana

```
3 public class Sarjana extends Mahasiswa implements ICumlaude {
4     public Sarjana(String nama) {
5         super(nama);
6     }
7
8     @Override
9     public void lulus() {
10        // TODO Auto-generated method stub
11        kuliahDiKampus();
12        System.out.println(x: "Aku sudah menyelesaikan SKRIPSI");
13    }
14
15    @Override
16    public void meraihIPKtinggi() {
17        // TODO Auto-generated method stub
18        System.out.println(x: "IPK-ku lebih dari 3,51");
19    }
```

b. PascaSarjana

```
3 public class PascaSarjana extends Mahasiswa implements ICumlaude {
4     public PascaSarjana(String nama) {
5         super(nama);
6     }
7
8     @Override
9     public void lulus() {
10        // TODO Auto-generated method stub
11        kuliahDiKampus();
12        System.out.println(x: "Aku sudah menyelesaikan TESIS");
13    }
14
15    @Override
16    public void meraihIPKtinggi() {
17        // TODO Auto-generated method stub
18        System.out.println(x: "IPK-ku lebih dari 3,71");
19    }
```

c. Output

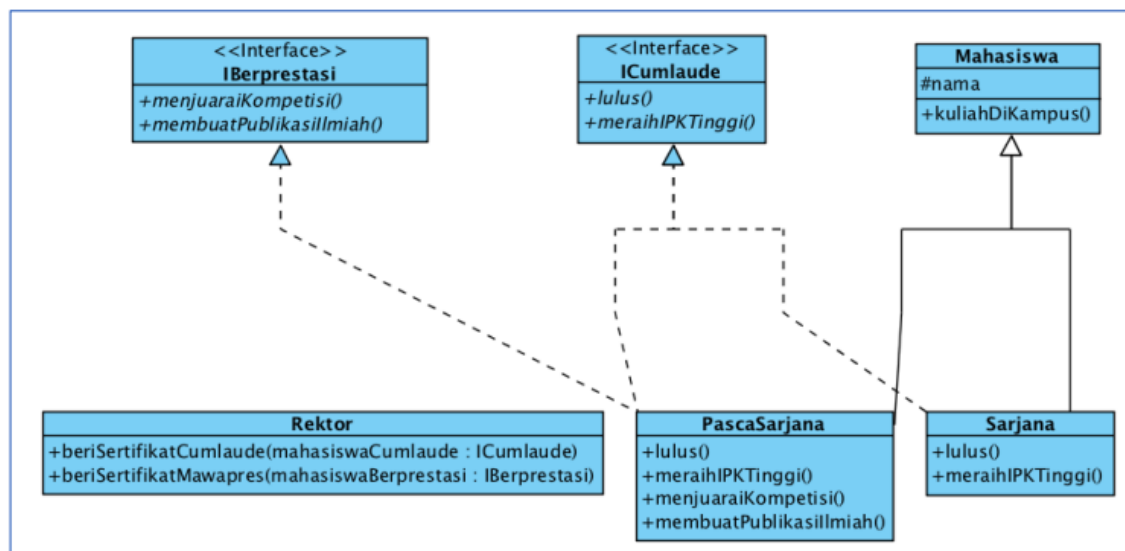

```

er 3\5. Pemrograman Berbasis Objek\Praktikum\
Saya REKTOR, memberikan sertifikat cumlaude
Selamat! silahkan perkenalkan diri Anda..
Aku mahasiswa, namaku Teguh
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 Saputro
Aku berkuliah di kampus.
Aku sudah menyelesaikan TESIS
IPK-ku lebih dari 3,71
-----
PS C:\Users\Halim\Downloads\POLINEMA\Semester

```

PRAKTIKUM 2. MUTIPLE INTERFACES IMPLEMENTATION

Pada percobaan kali ini kita akan memodifikasi program yang telah dibuat pada Percobaan 2 sehingga pada program tersebut nantinya akan terdapat sebuah class yang meng-*implements* lebih dari 1 *interface*.



Bayangkan pada skenario sebelumnya, dimana seorang rektor juga akan **beriSertifikatMawapres()** pada sebuah acara wisuda. Mahasiswa yang berhak menerima penghargaan tersebut tentunya adalah mahasiswa yang berprestasi, dimana kriteria prestasi di sini berbeda antara mahasiswa Sarjana dengan mahasiswa Pasca Sarjana. Pada percobaan ini, kita akan menentukan kriteria prestasi yaitu: harus **menjuaraiKompetisi()** dan **membuatPublikasiIlmiah()**.

1. Pada package yang sama dengan package pada Percobaan Sebelumnya, tambahkan sebuah interface baru Bernama IBerprestasi. Tambahkan baris kode seperti berikut didalamnya.

```
src > interfaceLatihan > J IBerprestasi.java > IBerprestasi > menjuaraiK
1  package interfaceLatihan;
2
3  public interface IBerprestasi {
4      public abstract void menjuaraiKompetisi();
5
6      public abstract void membuatPublikasiIlmiah();
7  }
8
```

2. Selanjutnya, modifikasilah class PascaSarjana dengan menambahkan interfave baru IBerprestasi dibelakang kata kunci implements. Lalu dengan cara yang sama seperti sebelumnya, kliklah ikon lampu peringatan untuk meng-generate semua method abstract dari interface pada class PascaSarjana.

```
src > interfaceLatihan > J PascaSarjana.java > PascaSarjana
1  package interfaceLatihan;
2
3  public class PascaSarjana extends Mahasiswa implements ICumlaude, IBerprestasi {
4      public PascaSarjana(String nama) {
```

3. Modifikasilah method yang telah di-generate oleh NetBeans menjadi seperti berikut

```
21  @Override
22  public void menjuaraiKompetisi() {
23      // TODO Auto-generated method stub
24      System.out.println(x: "Saya telah menjuarai kompetisi INTERNASIONAL");
25  }
26
27  @Override
28  public void membuatPublikasiIlmiah() {
29      // TODO Auto-generated method stub
30      System.out.println(x: "Saya menerbitkan artikel di jurnal INTERNASIONAL");
31  }
```

4. Tambahkan method beriSertifikatMawapres() dengan baris kode seperti di bawah, pada class Rektor.

```
14  public void beriSertifikatMawapres(IBerprestasi mahasiswa) {
15      System.out.println(x: "Saya REKTOR, memberikan sertifikat MAWAPRES.");
16      System.out.println(x: "Selamat! Bagaimana Anda bisa berprestasi?");
17
18      mahasiswa.menjuaraiKompetisi();
19      mahasiswa.membuatPublikasiIlmiah();
20
21      System.out.println(x: "-----");
22  }
```

5. Terakhir, modifikasilah method `main()` pada class `MultipleInterfaceMain` anda. Comment-lah semua baris yang terdapat method `beriSertifikatCumlaude()`, lalu tambahkan baris kode baru seperti pada gambar dibawah ini.

```
src > J MultipleInterfaceMain.java > ...
1  import interfaceLatihan.PascaSarjana;
2  import interfaceLatihan.Rektor;
3  import interfaceLatihan.Sarjana;
4
5  public class MultipleInterfaceMain {
    Run | Debug
6      public static void main(String[] args) {
7          Rektor pakRektor = new Rektor();
8
9          Sarjana sarjanaCum = new Sarjana(nama: "Halim");
10         PascaSarjana masterCum = new PascaSarjana(nama: "Teguh");
11
12         pakRektor.beriSertifikatMawapres(sarjanaCum);
13         pakRektor.beriSertifikatMawapres(masterCum);
14     }
15 }
```

6. akan terdapat error pada Langkah 5, sehingga program tidak dapat dieksekusi. Perbaikilah kode programnya, sehingga hasil eksekusi menjadi sama seperti pada screenshot dibawah ini.

```
Output - PBOMinggu10 (run)

run:
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: 0 seconds)
```

```
src > J MultipleInterfaceMain.java > MultipleInterfaceMain > main(String[])
1  import interfaceLatihan.PascaSarjana;
2  import interfaceLatihan.Rektor;
3  import interfaceLatihan.Sarjana;
4
5  public class MultipleInterfaceMain {
    Run | Debug
6      public static void main(String[] args) {
7          Rektor pakRektor = new Rektor();
8
9          Sarjana sarjanaCum = new Sarjana(nama: "Halim");
10         PascaSarjana masterCum = new PascaSarjana(nama: "Teguh");
11
12         // pakRektor.beriSertifikatMawapres(sarjanaCum);
13         pakRektor.beriSertifikatMawapres(masterCum);
14     }
15 }
16
```

```
face\jobsheet11\bin' 'MultipleInterfaceMain'
Saya REKTOR, memberikan sertifikat MAWAPRES.
Selamat! Bagaimana Anda bisa berprestasi?
Saya telah menjuarai kompetisi INTERNASIONAL
Saya menerbitkan artikel di jurnal INTERNASIONAL
-----
PS C:\Users\Halim\Downloads\POLINEMA\Semester 3\5.
```

PERTANYAAN PRAKTIKUM 2

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

Diganti : protected	←	<code>public</code>	<code>abstract void menjuaraiKompetisi();</code>
Diganti : private	←	<code>public</code>	<code>abstract void membuatPublikasiIlmiah();</code>

```
}
}
```

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

Jawab:

```

src > interfaceLatihan > IBerprestasi.java > ...
1  package interfaceLatihan;
2
3  public interface IBerprestasi {
4      protected abstract void menjuaraiKompetisi();
5
6      private abstract void membuatPublikasiIlmiah();
7  }
8

```

Keterangan: terjadi Error karena abstrak tidak bisa menggunakan acces modifier private karena pasti class abstrak tersebut akan di panggil diluar class tersebut dan abstrak tidak dianjurkan menggunakan acces modifier protected dikarenakan access modifier protected hanya dapat dilihat oleh clas itu sendiri, class yang ada di package itu sendiri, dan class yang berada pada subclass itu sendiri, jika menggunakan protected pada abstract, ini akan membuat program bingung akan mewariskan ke dalam packagenya atau ke bagian subclassnya. Jika keduanya bisa diwariskan, itu termasuk access modifier public.

- 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?

Jawab: terjadi error karena class IBerprestasi adalah class interface yang merupakan penghubung antar objek. class interface tidak dapat dibuat objek menggunakan new karena interface bersifat abstrak.

- Apabila Sarjana Berprestasi harus menjuarai kompetisi Nasional dan menerbitkan artike di jurnal Nasional, maka modifikasilah class-class yang terkait pada aplikasi anda agar di class Program objek pak Rektor dapat memberikan sertifikat mawapres pada objek sarjana Cumlaude.

Jawab:

```

Output - PBOMinggu10 (run)

run:
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)

```

a. Sarjana

```
package interfaceLatihan;

public class Sarjana extends Mahasiswa implements ICumlaude, IBerprestasi {
    public Sarjana(String nama) {
        super(nama);
    }

    @Override
    public void lulus() {
        // TODO Auto-generated method stub
        kuliahDiKampus();
        System.out.println("Aku sudah menyelesaikan SKRIPSI");
    }

    @Override
    public void meraihIPKTinggi() {
        // TODO Auto-generated method stub
        System.out.println("IPK-ku lebih dari 3,51");
    }

    @Override
    public void menjuaraiKompetisi() {
        // TODO Auto-generated method stub
        System.out.println("Saya telah menjuarai kompetisi NASIONAL");
    }

    @Override
    public void membuatPublikasiIlmiah() {
        // TODO Auto-generated method stub
        System.out.println("Saya menerbitkan artikel di jurnal NASIONAL");
    }
}
```

b. Output

```
in'
Saya REKTOR, memberikan sertifikat MAWAPRES.
Selamat! Bagaimana Anda bisa berprestasi?
Saya telah menjuarai kompetisi NASIONAL
Saya menerbitkan artikel di jurnal Nasional
-----
Selamat! Bagaimana Anda bisa berprestasi?
Saya telah menjuarai kompetisi INTERNASIONAL
Saya menerbitkan artikel di jurnal INTERNASIONAL
-----
```

LINK GITHUB: <https://github.com/HalimTeguh/Praktikum/tree/master/11.%20Interface>