PRAKTIKUM PBO JOBSHEET 11
INTERFACE DALAM OOP JAVA
PERTEMUAN 11

MARGARETHA VIOLINA PUTRI P.
2141762056 / 11
SIB-2F

POLITEKNIK NEGERI MALANG

2022

Daftar Isi

DAFTAR ISI **2**

1. KOMPETENSI **3**

2. PRAKTIKUM **3**

PERCOBAAN 1 - INTERFACE 3

PERTANYAAN PERCOBAAN 1 ..... *7*

PERCOBAAN 2 - MULTIPLE INTERFACES IMPLEMENTATION 9

PERTANYAAN PERCOBAAN 2 ..... *11*

4. TUGAS **13**

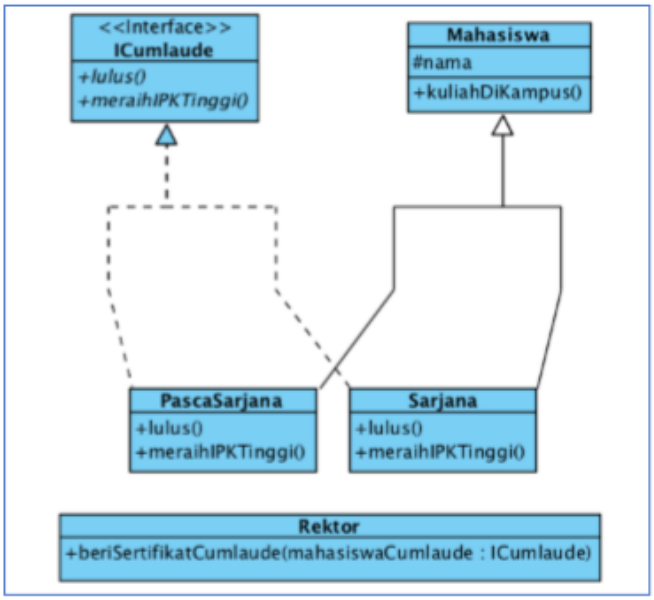
1. Kompetensi

Setelah menyelesaikan lembar kerja ini mahasiswa diharapkan mampu:
1. Menjelaskan maksud dan tujuan penggunaan Interface;
2. Menerapkan Interface di dalam pembuatan program

2. Praktikum

Percobaan 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.

Pada percobaan ini kita akan mencoba menerjemahkan skenario di atas ke dalam sebuah aplikasi sederhana yang memanfaatkan interface.

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().**

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

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

```
public class Mahasiswa {
    protected String nama;

    public Mahasiswa(String nama){
        this.nama = nama;
    }
    public void kuliahDiKampus(){
        System.out.println("Aku mahasiswa, namaku "+ this.nama);
        System.out.println( x: "Aku berkuliah di kampus. ");
    }
}
```

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.

6. Selanjutnya sesuaikan isi dari method **lulus()** dan **meraihIPKTinggi()** agar sama dengan baris kode di bawah.

```
public class Sarjana extends Mahasiswa implements ICumlaude{
    public Sarjana (String nama){
        super(nama);
    }
    @Override
    public void lulus(){
        System.out.println( x: "Aku sudah menyelesaikan SKRIPSI");
    }

    @Override
    public void meraihIPKTinggi() {
        System.out.println( x: "IPK-ku lebih dari 3,51");
    }
}
```

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** maka ia harus meng-**implements** interface **ICumlaude**.

7. Kemudian dengan **cara yang sama** buatlah class baru bernama **PascaSarjana** dengan baris kode seperti di bawah ini.

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

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

```
public class Rektor {
    public void beriSertifikatCumlaude(ICumlaude mahasiswa){
        System.out.println( x: "Saya REKTOR,memberikan sertifikat cumlaude");
        System.out.println( x: "Selamat! silahkan perkenalkan diri Anda..");

        mahasiswa.lulus();
        mahasiswa.meraihIPKTinggi();

        System.out.println( x: "----------------------------------------");
    }
}
```

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

```
5   package Jobsheet11_Interface;
    import Jobsheet11_Interface.ICumlaude;
    import Jobsheet11_Interface.Mahasiswa;
    import Jobsheet11_Interface.PascaSarjana;
    import Jobsheet11_Interface.Rektor;
    import Jobsheet11_Interface.Sarjana;
11
12  /**
13   *
14   * @author User
15   */
16  public class interfaceMain {
17      public static void main(String[] args){
18          Rektor pakrektor = new Rektor();
19
20          Mahasiswa mhsBiasa = new Mahasiswa( nama: "Charlie");
21          Sarjana sarjanaCumlaude = new Sarjana( nama: "Dini");
22          PascaSarjana masterCumlaude = new PascaSarjana( nama: "Elok");
23
24          pakrektor.beriSertifikatCumlaude(mhsBiasa);
25          pakrektor.beriSertifikatCumlaude( mahasiswa: sarjanaCumlaude);
26          pakrektor.beriSertifikatCumlaude( mahasiswa: masterCumlaude);
27
28      }
29  }
```

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:

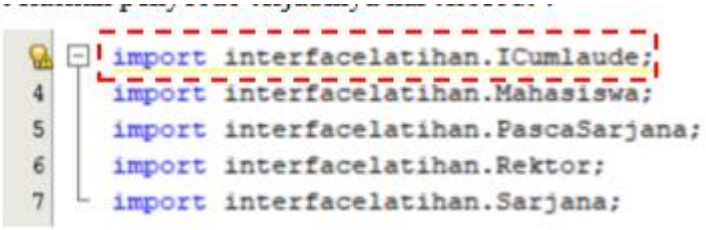
Jawab : Comment baris ke 24 agar tidak erorr

```
24  //          pakrektor.beriSertifikatCumlaude(mhsBiasa);
25          pakrektor.beriSertifikatCumlaude( mahasiswa: sarjanaCumlaude);
26          pakrektor.beriSertifikatCumlaude( mahasiswa: masterCumlaude);
27
28      }
29  }
30
```

```
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ Prak_PBO ---
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 SUCCESS
------------------------------------------------------------------------
Total time:  3.129 s
```

PERTANYAAN PERCOBAAN 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 mengimportkan dari package yang sama

2. Pada langkah ke 9, pada baris program ke 3. Apa yang terjadi jika script tersebut dihilangkan? Jelaskan menurut pemahaman anda.

Jawab : Tidak masalah jika baris kode import dari package yang sama diilangkan, program masih tetap akan berjalan. Kecuali class yang kita butuhkan dalam program berada diluar package yang berbeda, kita perlu gunakan import nama package dan class yang dituju.

3. Mengapa pada langkah nomor 9 terjadi error? Jelaskan!

Jawab : Karena class mahasiswa belum mengimplementasikan class interface yaitu ICumlaude. Jika mau tidak erorr maka class mahasiswa haruslah dirubah dengan menambahkan implements ICumlaude.

4. Dapatkah method **kuliahDiKampus()** dipanggil dari objek **sarjanaCumlaude** di class **Program**? Mengapa demikian?

Jawab : Bisa karena sarjanaCumlaude merupakan sub clas dari mahasiswa jadi method kuliahDiKampus dapat dipanggil.

5. Dapatkah method kuliahDiKampus() dipanggil dari parameter mahasiswa di method beriSertifikatCumlaude() pada class Rektor? Mengapa demikian?

Jawab : Tidak bisa karena class Rektor bukan subclass dari Mahasiswa sehingga method kuliahDiKampus tidak bisa dipanggil.

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

Jawab :

Tambahkan juga method kuliahDiKampus di interface ICumlaude

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

Ubah method beriSertifikatCumlaude() di class Rektor dengan menambahkan/memanggil method kuliahDiKampus :

```
public class Rektor {
    public void beriSertifikatCumlaude(ICumlaude mahasiswa){
        System.out.println( x: "Saya REKTOR,memberikan sertifikat cumlaude");
        System.out.println( x: "Selamat! silahkan perkenalkan diri Anda..");

        mahasiswa.kuliahDiKampus();
        mahasiswa.lulus();
        mahasiswa.meraihIPKTinggi();

        System.out.println( x: "----------------------------------------");
    }
}
```

Implementasikan method abstract kuliahDiKampus di Sarjana dan PascaSarjana seperti dibawah ini

```
    @Override
    public void kuliahDiKampus() {
        super.kuliahDiKampus();
    }
}
```

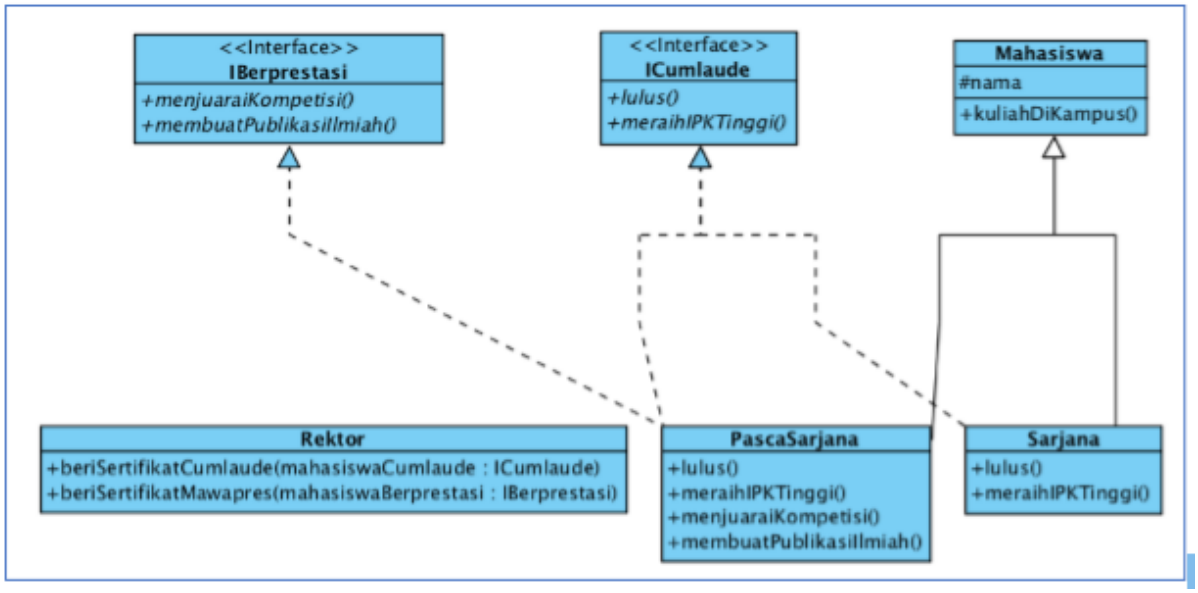
Isi body dengan key super yang berguna untuk memanggil method dari kelas induk yaitu method kuliahDiKampus class mahasiswa

Output :

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Prak_PBO ---
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 SUCCESS
------------------------------------------------------------------------
```

Percobaan 2 - Multiple 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 **memberiSertifikatMawapres**() 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 yang bernama **IBerprestasi**. Tambahkan baris kode seperti berikut didalamnya.

```
10      */
11      public interface IBerprestasi {
12          public abstract void menjuaraiKompetisi();
13          public abstract void membuatPublikasiIlmiah();
14      }
15
```

2. Selanjutnya, **modifikasilah** class **PascaSarjana** dengan menambahkan interface 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 **IBerprestasi** pada class **PascaSarjana**.

```
10      */
11      public class PascaSarjana extends Mahasiswa implements ICumlaude, IBerprestasi{
12          public PascaSarjana(String nama){
```

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

```
30      @Override
31      public void menjuaraiKompetisi() {
32          System.out.println("Saya telah menjuarai kompetisi INTERNASIONAL");
33      }
34
35      @Override
36      public void membuatPublikasiIlmiah() {
37          System.out.println("Saya menerbitkan artikel di jurnal INTERNASIONAL");
38      }
39  }
```

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

```
public void beriSertifikatMawapres(IBerprestasi mahasiswa){
    System.out.println("Saya REKTOR,memberikan sertifikat MAWAPRES.");
    System.out.println("Selamat! Bagaimana Anda bisa berprestasi?");

    mahasiswa.menjuaraiKompetisi();
    mahasiswa.membuatPublikasiIlmiah();

    System.out.println("----------------------------------------");
}
```

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 di bawah ini.

```
16  public class MultipleInterfaceMul {
17
18      /**
19       * @param args the command line arguments
20       */
21      public static void main(String[] args) {
22          Rektor pakRektor = new Rektor();
23
24          Sarjana sarjanaCum = new Sarjana("Dini");
25          PascaSarjana masterCum = new PascaSarjana("Elok");
26
            pakRektor.beriSertifikatMawapres(sarjanaCum);
28          pakRektor.beriSertifikatMawapres(masterCum);
29      }
```

6. Akan terdapat *error* pada langkah-5, sehingga program tidak dapat dieksekusi. Perbaikilah kode programmnya, sehingga hasil eksekusi menjad **sama** seperti pada *screenshot* di bawah ini.

Jawab : Lakukan comment pada baris berikut di class MultipleInterfaceMul

```
public class MultipleInterfaceMul {
    public static void main(String[] args){
        Rektor pakrektor = new Rektor();

        Sarjana sarjanaCum = new Sarjana( nama: "Dini");
        PascaSarjana masterCum = new PascaSarjana( nama: "Elok");

        //pakrektor.beriSertifikatMawapres(sarjanaCum);
        pakrektor.beriSertifikatMawapres( mahasiswa: masterCum);

    }
}
```

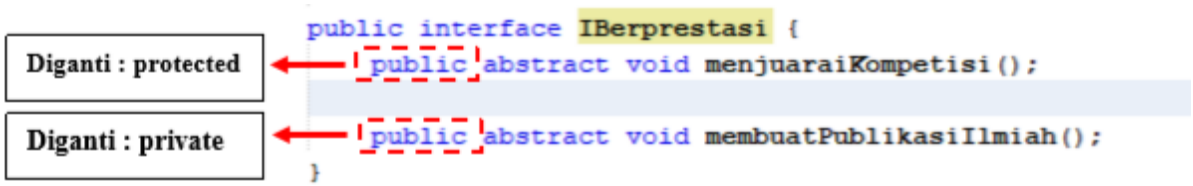
Output :

```
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Prak_PBO ---
Saya REKTOR,memberikan sertifikat MAWAPRES.
Selamat! Bagaimana Anda bisa berprestasi?
Saya telah menjuarai kompetisi INTERNASIONAL
Saya menerbitkan artikel di jurnal INTERNASIONAL
---------------------------------------
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  3.646 s
Finished at: 2022-11-26T14:20:58+07:00
------------------------------------------------------------------------
```

PERTANYAAN PERCOBAAN 2

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)

Jawab : Hasilnya akan error, juga error bila kita memberikan konstanta pada class interfcae karena class interface harus public sesuai aturan dan class interface akna dipakai di beberapa class sehingga tidak bisa private/protected.

```
10  */
    public interface IBerprestasi {
        protected abstract void menjuaraiKompetisi();
        private abstract void membuatPublikasiIlmiah();
14  }
```

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 ?

Jawab : Karena di class interface tidak bisa membuat objek. Interface tidak bisa di instanisasi objek dengan kata new, tetapi bisa di instanisasi melalui class yang meng-implement-nya

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**

   Jawab :
   Tambahkan implementasi IBerprestasi di class Sarjana

```
    */
public class Sarjana extends Mahasiswa implements ICumlaude, IBerprestasi{
    public Sarjana (String nama){
```

   Jangan lupa juga untuk mengimplementasikan abstract method yang ada di class interface IBerprestasi

```
    @Override
    public void menjuaraiKompetisi() {
        System.out.println( x: "Saya telah menjuarai kompetisi NASIONAL");
    }

    @Override
    public void membuatPublikasiIlmiah() {
        System.out.println( x: "Saya menerbitkan artikel di jurnal NASIONAL");
    }
}
```

   Lalu buka comment pada baris yang sbeelumnya di comment di class MultipleInterfaceMul seperti dibawah ini

```
*/
public class MultipleInterfaceMul {
    public static void main(String[] args){
        Rektor pakrektor = new Rektor();

        Sarjana sarjanaCum = new Sarjana( nama: "Dini");
        PascaSarjana masterCum = new PascaSarjana( nama: "Elok");

        pakrektor.beriSertifikatMawapres( mahasiswa: sarjanaCum);
        pakrektor.beriSertifikatMawapres( mahasiswa: masterCum);

    }
}
```

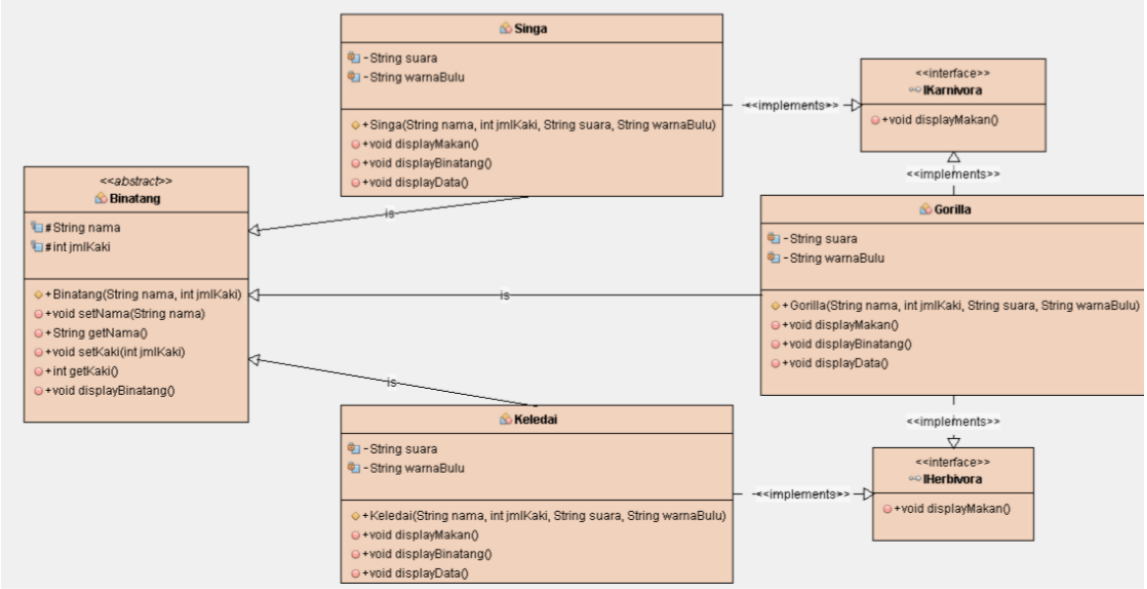
Output

```
--------------------------------[ jar ]---------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Prak_PBO ---
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 SUCCESS
------------------------------------------------------------------------
Total time:  3.688 s
Finished at: 2022-11-26T14:26:06+07:00
------------------------------------------------------------------------
```

4. Tugas

Terdapat sebuah UML diagram dari sistem yang mengidentifikasi tentang karakteristik dan jenis makanan dari beberapa binatang, seperti gambar dibawah ini. **Singa** hanya makan jenis daging yang tergolong sebagai jenis **Karnivora**, sedangkan **Keledai** termasuk binatang **Herbivora** karena hanya mengkonsumsi jenis tumbuhan. Sedangkan untuk **Gorilla** dapat mengkonsumsi makanan dari kedua jenis tersebut, dapat disebut dengan **omnivora** dan dapat dikategorikan sebagai **kombinasi dari Herbivora dan Karnivora**. Dalam UML tersebut, juga terdapat beberapa karakteristik dari binatang seperti **nama, suara, warnabulu dan jumlah kaki.**



Dengan penjelasan kasus dan UML diatas, implementasikan sebuah sistem pada Java dengan output seperti gambar dibawah ini. **Hasil tugas dapat dibentuk laporan dengan capture script code beserta penjelasannya. (Sertakan juga hasil UML dari sistem anda).**

Jawab :

Class Abstract Binatang

```
public abstract class Binatang {
    protected String nama;
    protected int jmlKaki;

    Binatang (String nama, int jmlKaki){
        this.nama = nama;
        this.jmlKaki = jmlKaki;
    }
    public void setNama(String nama){
        this.nama = nama;
    }
    public String getNama(){
        return nama;
    }
    public void setKaki(int jmlKaki){
        this.jmlKaki = jmlKaki;
    }
    public int getKaki(){
        return jmlKaki;
    }
    public void displayBinatang(){
        System.out.println("Nama        : "+this.nama);
        System.out.println("Jumlah Kaki: "+this.jmlKaki);
    }
}
```

Class Interface Herbivora

```
10  */
    public interface Herbivora {
        public abstract void displayMakan();
13  }
14
```

Class Interface Karnivora

```
public interface Karnivora {
    public abstract void displayMakan();
}
```

Class Singa

```
11  public class Singa extends Binatang implements Karnivora{
        private String suara, warnaBulu;
13
14
15      public Singa(String nama, int jmlKaki, String suara, String warnaBulu) {
16          super(nama, jmlKaki);
17          this.suara = suara;
18          this.warnaBulu = warnaBulu;
19      }
20
21      @Override
        public void displayMakan() {
23          System.out.println( x: "Makanan    : Daging");
24      }
        public void displayBinatang() {
26          System.out.println( x: "Jenis      : Karnivora");
27      }
28      public void displayData() {
29          System.out.println( x: "---------Binatang---------");
30          this.displayBinatang();
31          this.displayMakan();
32          super.displayBinatang();
33          System.out.println("Suara       :"+ this.suara);
            System.out.println("Warna Bulu :"+ this.warnaBulu);
35      }
36  }
```

Class Keledai

```
11  public class Keledai extends Binatang implements Herbivora{
        private String suara, warnaBulu;
13
14
15      public Keledai(String nama, int jmlKaki, String suara, String warnaBulu) {
16          super(nama, jmlKaki);
17          this.suara = suara;
18          this.warnaBulu = warnaBulu;
19      }
20
21      @Override
        public void displayMakan() {
23          System.out.println( x: "Makanan    : Tumbuhan");
24      }
        public void displayBinatang() {
26          System.out.println( x: "Jenis      : Herbivora");
27      }
28      public void displayData() {
29          System.out.println( x: "---------Binatang---------");
30          this.displayBinatang();
31          this.displayMakan();
32          super.displayBinatang();
33          System.out.println("Suara       :"+ this.suara);
            System.out.println("Warna Bulu :"+ this.warnaBulu);
35      }
36  }
```

Class Gorilla

```
11  public class Gorilla extends Binatang implements Herbivora, Karnivora{
        private String suara, warnaBulu;
13
14      public Gorilla(String nama, int jmlKaki, String suara, String warnaBulu) {
15          super(nama, jmlKaki);
16          this.suara = suara;
17          this.warnaBulu = warnaBulu;
18      }
19
20      @Override
        public void displayMakan() {
22          System.out.println( x: "Makanan    : Daging + Tumbuhan");
23      }
        public void displayBinatang() {
25          System.out.println( x: "Jenis      : Karnivora + Herbivora");
26      }
27      public void displayData() {
28          System.out.println( x: "---------Binatang---------");
29          this.displayBinatang();
30          this.displayMakan();
31          super.displayBinatang();
32          System.out.println("Suara       :"+ this.suara);
            System.out.println("Warna Bulu :"+ this.warnaBulu);
34      }
35  }
```

Output

INTERFACE

```
--- exec-maven-plugin:3.0.0:exec (default-cli)
---------Binatang---------
Jenis      : Herbivora
Makanan    : Tumbuhan
Nama       : Keledai
Jumlah Kaki: 4
Suara      :Hehehe
Warna Bulu :Abu-Abu
---------Binatang---------
Jenis      : Karnivora + Herbivora
Makanan    : Daging + Tumbuhan
Nama       : Gorilla
Jumlah Kaki: 4
Suara      :Haaum Hauum
Warna Bulu :Hitam Manis
---------Binatang---------
Jenis      : Karnivora
Makanan    : Daging
Nama       : Singa
Jumlah Kaki: 4
Suara      :Roaar Roaarr
Warna Bulu :Cokelat
------------------------------------------------
BUILD SUCCESS
------------------------------------------------
Total time:  3.210 s
Finished at: 2022-11-25T00:12:52+07:00
```