



NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

## I. Abstract Class

1. Buatlah sebuah project baru di NetBeans dengan nama PBOMinggu10
1. Pada package pbominggu10, tambahkan package baru dengan cara klik kanan nama package New Java Package...
2. Beri nama package tersebut dengan nama abstractclass. Semua class yang dibuat pada percobaan 1 ini diletakkan pada package yang sama, yaitu package abstractclass ini,
3. Pada package baru tersebut tambahkan class baru.
4. Beri nama class baru tersebut, yaitu class Hewan.

```
public abstract class Hewan {  
    private int umur;  
    protected Hewan() {  
        this.umur = 0;  
    }  
    public void bertambahUmur() {  
        this.umur += 1;  
    }  
    public abstract void bergerak();  
}
```

5. Pada class Hewan tersebut, ketikkan kode berikut ini.  
Class Hewan tersebut adalah class abstract berisi property dan method biasa, ditambah sebuah method abstract bernama bergerak(). Method tersebut didepannya terdapat kata kunci abstract dan tidak memiliki badan fungsi. Method ini nantinya akan di-override oleh class mana saja yang menjadi class turunan dari class Hewan tersebut.
6. Dengan cara yang sama, buatlah class dengan nama Kucing yang meng-extend class Hewan. Di dalam class Kucing tersebut, setelah Anda menuliskan kode seperti di bawah, maka akan muncul ikon lampu peringatan. Klik lampu tersebut dan kemudian pilih implement all abstract methods.
7. Maka akan secara otomatis dibuatkan fungsi yang meng-override fungsi abstract bergerak() yang ada pada class hewan.
8. Ubahlah badan fungsi tersebut dengan mengganti kode didalamnya menjadi seperti berikut.

```
public class Kucing extends Hewan {  
    @Override  
    public void bergerak() {  
        System.out.println("Berjalan dengan KAKI, \nTap..tap..\n");  
    }  
}
```

9. Dengan cara yang sama seperti ketika Anda membuat class Kucing, buatlah class Hewan baru bernama Ikan dan buatlah kodenya seperti pada gambar dibawah.

```
public class Ikan extends Hewan {  
    public void bergerak() {  
        System.out.println("Berenang dengan SIRIP. \nwush..wush..\n");  
    }  
}
```

10. Selanjutnya, buatlah class biasa baru yang bernama class Orang. Class ini adalah class yang menjadi pengguna dari class abstract Hewan yang sudah dibuat sebelumnya. Ketikkan pada class



NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

Orang tersebut, baris-baris kode seperti di bawah.

```
public class Orang {  
    private String nama;  
    private Hewan hewanPeliharaan;  
    public Orang(String nama){  
        this.nama = nama;  
    }  
    public void peliharaHewan(Hewan hewanPeliharaan){  
        this.hewanPeliharaan = hewanPeliharaan;  
    }  
    public void ajakPeliharaanJalan(){  
        System.out.println("Nama : "+nama);  
        System.out.println("Hewan peliharaanku berjalan dengan cara: ");  
        this.hewanPeliharaan.bergerak();  
        System.out.println("-----");  
    }  
}
```

11. Terakhir, buatlah sebuah Main Class baru di dalam package yang sama. Beri nama class baru tersebut dengan nama class Program. Ketikkan didalamnya seperti kode di bawah ini.

```
public class Program {  
    public static void main(String[] args) {  
        Kucing kucingKampung = new Kucing();  
        Ikan lumbaLumba = new Ikan();  
  
        Orang ani = new Orang("Ani");  
        Orang budi = new Orang("Budi");  
  
        ani.peliharaHewan(kucingKampung);  
        budi.peliharaHewan(lumbaLumba);  
        ani.ajakPeliharaanJalan();  
        budi.ajakPeliharaanJalan();  
    }  
}
```

12. Jalankan class tersebut dengan cara klik kanan pada class Program kemudian pilih Run File

13. Perhatikan dan amati hasilnya!

```
run:  
Nama : Ani  
Hewan peliharaanku berjalan dengan cara:  
Berjalan dengan KAKI, "Tap..tap.."  
-----  
Nama : Budi  
Hewan peliharaanku berjalan dengan cara:  
Berenang dengan SIRIP. "wush..wush.."  
-----  
BUILD SUCCESSFUL (total time: 0 seconds)
```



NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

#### 14. Pertanyaan diskusi:

Bolehkah apabila sebuah class yang meng-extend suatu abstract class tidak mengimplementasikan method abstract yang ada di class induknya? Buktikan!

**Jawab :** tidak boleh, jika abstract methodnya tidak diimplementasikan maka akan menghasilkan error.

```
package pbominggul0.abstractclass;

/**
 *
 * @author LENOVO
 */
public class Ikan extends Hewan {
}
```

Ikan is not abstract and does not override abstract method bergerak() in Hewan  
(Alt-Enter shows hints)

Disini class ikan mengextend kelas hewan yang memiliki method abstract bergerak tapi tidak mengimplementasikannya, maka akan muncul eror seperti ini.

## II. Interface

### A. Percobaan 1

- 1) Pada project latihan PBOMinggu10, buat sebuah package baru bernama interfacelatihan. Kemudian klik
- 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().

```
package pbominggul0.interfaceLatihan;

/**
 *
 * @author LENOVO
 */
public interface IBerprestasi {
    public abstract void menjuaraiKompetisi();
    public abstract void membuatPublikasiIlmiah();
}
```

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



NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

```
package pbominggul0.interfaceLatihan;

/**
 *
 * @author LENOVO
 */
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("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.

```
public class Sarjana extends Mahasiswa implements ICumlaude {

    public Sarjana(String nama) {
        super(nama);
    }

    public void lulus() {
        System.out.println("Aku sudah menyelesaikan SKIPSI");
    }

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

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





NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

```
public class PascaSarjana extends Mahasiswa implements ICumlaude{  
  
    public PascaSarjana(String nama) {  
        super(nama);  
    }  
  
    @Override  
    public void lulus() {  
        System.out.println("Aku sudah menyelesaikan THESIS");  
    }  
  
    @Override  
    public void meraihIPKTinggi() {  
        System.out.println("IPK-ku lebih dari 3,71");  
    }  
}
```

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("Saya REKTOR, memberikan sertifikat cumlaude.");  
        System.out.println("Selamat! silahkan perkenalkan diri Anda..");  
        mahasiswa.lulus();  
        mahasiswa.meraihIPKTinggi();  
        System.out.println("-----");  
    }  
}
```

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

```
public class interfacemain {  
    public static void main(String[] args) {  
        Rektor pakrektor = new Rektor();  
        Mahasiswa mhsBiasa = new Mahasiswa("Charlie");  
        Sarjana sarjanaCumlaude = new Sarjana("Dini");  
        PascaSarjana masterCumlaude = new PascaSarjana("Elok");  
        pakrektor.beriSertifikatCumlaude(mhsBiasa);  
        pakrektor.beriSertifikatCumlaude(sarjanaCumlaude);  
        pakrektor.beriSertifikatCumlaude(masterCumlaude);  
    }  
}
```

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:



NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

```
public class interfacemain {  
    public static void main(String[] args) {  
        Rektor pakrektor = new Rektor();  
        Sarjana sarjanaCumlaude = new Sarjana("Dini");  
        PascaSarjana masterCumlaude = new PascaSarjana("Elok");  
        pakrektor.beriSertifikatCumlaude(sarjanaCumlaude);  
        pakrektor.beriSertifikatCumlaude(masterCumlaude);  
    }  
}
```

```
Run:  
Saya REKTOR, memberikan sertifikat cumlaude.  
Selamat! silahkan perkenalkan diri Anda..  
Aku sudah menyelesaikan SKIPSI  
IPK-ku lebih dari 3,51  
-----  
Saya REKTOR, memberikan sertifikat cumlaude.  
Selamat! silahkan perkenalkan diri Anda..  
Aku sudah menyelesaikan THESIS  
IPK-ku lebih dari 3,71  
-----  
BUILD SUCCESSFUL (total time: 0 seconds)
```

## B. Percobaan 2

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

```
public interface IBerprestasi {  
    public abstract void menjuaraiKompetisi();  
    public abstract void membuatPublikasiIlmiah();  
}
```

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.



NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

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

```
public class PascaSarjana extends Mahasiswa implements ICumlaude, IBerprestasi {

    @Override
    public void menjuaraiKompetisi() {
        System.out.println("Saya telah menjuarai kompetisi INTERNASIONAL");
    }

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

    public PascaSarjana(String nama) {
        super(nama);
    }

    @Override
    public void lulus() {
        System.out.println("Aku sudah menyelesaikan THESIS");
    }

    @Override
    public void meraihIPKTinggi() {
        System.out.println("IPK-ku lebih dari 3,71");
    }

}
```

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

```
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.lulus();
        mahasiswa.meraihIPKTinggi();
        System.out.println("-----");
    }

    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("-----");
    }

}
```





NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

5) Terakhir, modifikasilah method main() pada class MultipleInterfaceMain Anda. Comment-lah semua baris

```
public class MultipleInterfaceMain {  
    public static void main(String[] args) {  
        Rektor pakrektor = new Rektor();  
        Sarjana sarjanaCum = new Sarjana("Dini");  
        PascaSarjana masterCum = new PascaSarjana("Elok");  
        pakrektor.beriSertifikatMawapres(sarjanaCum);  
        pakrektor.beriSertifikatMawapres(masterCum);  
    }  
}
```

yang terdapat method beriSertifikatCumlaude(), lalu tambahkan baris kode baru seperti pada gambar di bawah ini.

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

```
public class MultipleInterfaceMain {  
    public static void main(String[] args) {  
        Rektor pakrektor = new Rektor();  
        // Sarjana sarjanaCum = new Sarjana("Dini");  
        PascaSarjana masterCum = new PascaSarjana("Elok");  
        // pakrektor.beriSertifikatMawapres(sarjanaCum);  
        pakrektor.beriSertifikatMawapres(masterCum);  
    }  
}
```

```
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)
```





NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

## 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.

## OUTPUT :

```
run:
=====
binatang ini adalah SINGA
Singa adalah binatang pemakan DAGING
Nama binatang = SINGA
Jumlah kaki   =4
Suara         = RAWR..
Wana bulu    = Kuning
=====
=====
Ini adalah binatang KELEDAI
Keledai adalah binatang pemakan TUMBUHAN
Nama binatang = Keledai
Jumlah kaki   =4
Suara         = Oik..
Wana bulu     = Abu-abu
=====
=====
Ini adalah binatang GORILLA
Gorilla adalah binatang pemakan SEGALA
Nama binatang = Gorilla
Jumlah kaki   =2
Suara         = Huhahaha
Wana bulu     = Hitam
=====
BUILD SUCCESSFUL (total time: 0 seconds)
```



NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

#### MAIN :

```
public class Main {  
    public static void main(String[] args) {  
        Singa sg = new Singa("RAWR..", "Kuning", "SINGA", 4);  
        Keledai kd = new Keledai("Oik..", "Abu-abu", "Keledai", 4);  
        Gorilla gr = new Gorilla("Huhuhaha", "Hitam", "Gorilla", 2);  
  
        sg.displayBinatang();  
        sg.displayMakan();  
        sg.displayData();  
  
        kd.displayBinatang();  
        kd.displayMakan();  
        kd.displayData();  
  
        gr.displayBinatang();  
        gr.displayMakan();  
        gr.displayData();  
    }  
}
```

#### CLASS KELEDAI

```
public class Keledai extends Binatang implements IHerbivora {  
    private String suara;  
    private String warnaBulu;  
  
    public Keledai(String suara, String warnaBulu, String nama, int jmlKaki) {  
        super(nama, jmlKaki);  
        this.suara = suara;  
        this.warnaBulu = warnaBulu;  
    }  
  
    @Override  
    public void displayBinatang() {  
        System.out.println("=====");  
        System.out.println("Ini adalah binatang KELEDAI");  
    }  
  
    @Override  
    public void displayMakan() {  
        System.out.println("Keledai adalah binatang pemakan TUMBUHAN");  
    }  
  
    public void displayData() {  
        System.out.println("Nama binatang = " + nama);  
        System.out.println("Jumlah kaki    = " + jmlKaki);  
        System.out.println("Suara        = " + suara);  
        System.out.println("Wana bulu    = " + warnaBulu);  
        System.out.println("=====");  
    }  
}
```



NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

#### CLASS GORILA:

```
public class Gorilla extends Binatang implements IKarnivora,IHerbivora{
    private String suara;
    private String warnaBulu;

    public Gorilla(String suara, String warnaBulu, String nama, int jmlKaki) {
        super(nama, jmlKaki);
        this.suara = suara;
        this.warnaBulu = warnaBulu;
    }

    @Override
    public void displayBinatang() {
        System.out.println("=====");
        System.out.println("Ini adalah binatang GORILLA");
    }

    @Override
    public void displayMakan() {
        System.out.println("Gorilla adalah binatang pemakan SEGALA");
    }

    public void displayData() {
        System.out.println("Nama binatang = " + nama);
        System.out.println("Jumlah kaki    = " + jmlKaki);
        System.out.println("Suara        = " + suara);
        System.out.println("Wana bulu    = " + warnaBulu);
        System.out.println("");
        System.out.println("=====");
    }
}
```

#### CLASS SINGA :

```
public class Singa extends Binatang implements IKarnivora{
    private String suara;
    private String warnaBulu;

    public Singa(String suara, String warnaBulu, String nama, int jmlKaki) {
        super(nama, jmlKaki);
        this.suara = suara;
        this.warnaBulu = warnaBulu;
    }

    @Override
    public void displayBinatang() {
        System.out.println("=====");
        System.out.println("binatang ini adalah SINGA");
    }

    @Override
    public void displayMakan() {
        System.out.println("Singa adalah binatang pemakan DAGING");
    }

    public void displayData() {
        System.out.println("Nama binatang = " + nama);
        System.out.println("Jumlah kaki    = " + jmlKaki);
        System.out.println("Suara        = " + suara);
        System.out.println("Wana bulu    = " + warnaBulu);
        System.out.println("=====");
    }
}
```



NAMA :ANNISA AULIA NADHILA  
NIM :2041720023  
KELAS :TI -2C  
KETERANGAN :Abstract dan interface

### CLASS BINATANG

```
public abstract class Binatang {  
    protected String nama;  
    protected int jmlKaki;  
  
    public Binatang(String nama, int jmlKaki) {  
        this.nama = nama;  
        this.jmlKaki = jmlKaki;  
    }  
  
    public String getNama() {  
        return nama;  
    }  
  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
  
    public int getJmlKaki() {  
        return jmlKaki;  
    }  
  
    public void setJmlKaki(int jmlKaki) {  
        this.jmlKaki = jmlKaki;  
    }  
    public abstract void displayBinatang();  
}
```

### INTERFACE KARNIVORA :

```
public interface IKarnivora {  
    public void displayMakan();  
}
```

### INTERFACE HERBIVORA :

```
public interface IHerbivora {  
    public void displayMakan();  
}
```