

LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS OBJEK
JOBSHEET 10
(Abstract Class)



Oleh :

Debora Febriana P. A
TI2B / 08
1941720156

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

Percobaan 1

Class Hewan

```
package Jobsheet10.abstractclass;

/**
 *
 * @author DEBORA
 */
public abstract class hewan {
    private int umur;
    protected hewan() {
        this.umur=0;
    }
    public void bertambahUmur() {
        this.umur+=1;
    }
    public abstract void bergerak();
}
```

Class Kucing

```

    * @author DEBORA
    */
public class kucing extends hewan{
    @Override
    public void bergerak() {
        System.out.println("Berjalan dengan KAKI, \"tap..tap..\"");
    }
}
```

Class Ikan

```

    * @author DEBORA
    */
public class ikan extends hewan{
    @Override
    public void bergerak() {
        System.out.println("Berenang dengan SIRIP, \"wush..wush..\"");
    }
}
```

Class Orang

```
...
* @author DEBORA
*/
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 hewanPeliharaan(hewan hewanPeliharaan){
        this.hewanPeliharaan=hewanPeliharaan;
    }

    public void ajakPeliharaanJalanJalan(){
        System.out.println("Namaku "+this.nama);
        System.out.println("Hewan peliharaanku berjalan dengan cara: ");
        this.hewanPeliharaan.bergerak();
        System.out.println("-----");
    }
}
```

Class Program

```
...
* @author DEBORA
*/
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.ajakPeliharaanJalanJalan();
        budi.ajakPeliharaanJalanJalan();
    }
}
```

Output

```
compile-single:
run-single:
Namaku Ani
Hewan peliharaanku berjalan dengan cara:
Berjalan dengan KAKI, "tap..tap.."
-----
Namaku Budi
Hewan peliharaanku berjalan dengan cara:
Berenang dengan SIRIP, "wush..wush.."
-----
BUILD SUCCESSFUL (total time: 1 second)
```

Pertanyaan

1. Berikan penjelasan terkait tentang jalannya program diatas
2. Tunjukkan hasil kompilasi program dan berikan penjelasan singkat jika *method* **bergerak()** diubah menjadi *method abstract*!
3. Tunjukkan hasil kompilasi program dan berikan penjelasan singkat jika tidak dilakukan *overriding* terhadap *method* **bergerak()**
4. Tunjukkan hasil kompilasi program dan berikan penjelasan singkat jika *abstract method* **bergerak()** yang dideklarasikan dalam Class Ikan

Jawaban

1. Class Hewan adalah class abstarct. Class Hewan merupakan class parent dari class Kucing dan class Ikan. Class Hewan berisi Sedangkan class Orang adalah pengguna dari class abstract (class Hewan). Pada class Program, class ini menginstansiasikan objek Kucing dan Ikan ke objek hewanPeliharaan. Setelah itu, pada class Orang memanggil method ajakPeliharaanjalanJalan() yang dimana dalam method tersebut memanggil method bergerak() yang ada di dalam class Hewab. Class Hewan memanggil method bergerak() dari class turunannya sesuai dengan objek yang di inisialisasikan.
2. Terjadi error sebab class Kucing bukan class abstract dan merupakan class turunan dari class Hewan yang merupakan class abstract.

```

9      * @author DEBORA
10     */
11     public class kucing extends hewan{
12         @Override
13         public abstract void bergerak(){
14             System.out.println("Berjalan dengan KAKI, \"tap..tap..\"");
15         }
16     }
17
C:\Users\DEBORA\Documents\NetBeansProjects\Algoritma
public class kucing extends hewan{
C:\Users\DEBORA\Documents\NetBeansProjects\Algoritma
    public abstract void bergerak(){
2 errors
BUILD FAILED (total time: 1 second)

```

3. Hasil compile sama

```

11     public class kucing extends hewan{
12         // @Override
13         public void bergerak(){
14             System.out.println("Berjalan dengan KAKI, \"tap..tap..\"");
15         }
16     }
17
run-single:
Namaku Ani
Hewan peliharaanmu berjalan dengan cara:
Berjalan dengan KAKI, "tap..tap.."
-----
Namaku Budi
Hewan peliharaanmu berjalan dengan cara:
Berenang dengan SIRIP, "wush..wush.."
-----
BUILD SUCCESSFUL (total time: 1 second)

```

4. Tentunya terjadi error karena class Ikan extends pada class Hewan serta class Ikan itu sendiri bukanlah abstract class.

```

10     * @author DEBORA
11     */
12     public class ikan extends hewan{
13         @Override
14         public abstract void bergerak(){
15             System.out.println("Berenang dengan SIRIP, \"wush..wush..\"");
16         }
17     }
18
C:\Users\DEBORA\Documents\NetBeansProjects\Algoritma
public class ikan extends hewan{
C:\Users\DEBORA\Documents\NetBeansProjects\Algoritma
    public abstract void bergerak(){
2 errors
BUILD FAILED (total time: 1 second)

```