

**LAPORAN PRAKTIKUM**  
**MATA KULIAH PEMROGRAMAN BERBASIS OBJEK**

Dosen Pengampu : Vit Zuraida, S.Kom, M.Kom

**JOBSHEET - 8**



**Nama : M. Zidna Billah Faza**  
**NIM : 2341760030**  
**Prodi : D-IV Sistem Informasi Bisnis**

**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2024**

# ABSTRACT

## A. Percobaan 1

- 1) Buatlah project baru dengan nama Praktikum08 kemudian buat class baru dengan nama Hewan. Method bernapas dan bergerak tidak memiliki statement atau baris kode.

```
public class Hewan {  
  
    public double berat;  
    public String makanan;  
    public String habitat;  
  
    public Hewan(double berat, String makanan, String habitat) {  
        this.berat = berat;  
        this.makanan = makanan;  
        this.habitat = habitat;  
    }  
  
    public void bergerak() {  
  
    }  
  
    public void bernapas() {  
  
    }  
  
    public void cetakInfo() {  
        System.out.println("Berat      : " + this.berat);  
        System.out.println("Makanan   : " + this.makanan);  
        System.out.println("Habitat  : " + this.habitat);  
    }  
}
```

- 2) Buat class Lebah sebagai subclass dari class Hewan sebagai berikut

```
public class Lebah extends Hewan {  
  
    public String kasta;  
  
    public Lebah( String kasta, double berat, String makanan, String habitat) {  
        super(berat, makanan, habitat);  
        this.kasta = kasta;  
    }  
}
```

- 3) Buat class main dengan nama AbstractClassDemo lalu instansiasi objek dari class Hewan dan class Lebah. Run program kemudian amati hasilnya.

```
public class AbstractClassDemo {  
    Run | Debug  
    public static void main(String[] args) {  
  
        Hewan hewan1 = new Hewan(berat:10, makanan:"Rumput", habitat:"Savana");  
        hewan1.cetakInfo();  
        hewan1.bergerak();  
        hewan1.bernapas();  
  
        Lebah lebah1 = new Lebah(kasta:"Ratu", berat:0.05, makanan:"Nektar", habitat:"Hutan");  
        lebah1.cetakInfo();  
        lebah1.bergerak();  
        lebah1.bernapas();  
    }  
}
```

```
Berat      : 10.0  
Makanan    : Rumput  
Habitat    : Savana  
Berat      : 0.05  
Makanan    : Nektar  
Habitat    : Hutan
```

## B. Pertanyaan Percobaan 1 (extends)

- 1) Bagaimana hasil pada langkah 3? Apakah objek hewan1 dan lebah1 berhasil diinstansiasi?

Kedua obek berhasil dibuat.

- 2) Menurut Anda, mengapa tidak ada baris program pada method bergerak() dan bernapas() pada class Hewan()?

Karena isi dari method tersebut kosong.

- 3) Class Lebah tidak memiliki method bergerak(), bernapas(), dan cetakInfo(), mengapa tidak terjadi error pada AbstractClassDemo?

Karena method tersebut turunan dari class Hewan.

## C. Percobaan 2 (Pewarisan)

1) Ubah method bergerak dan bernapas menjadi abstract method.

```
public class Hewan {  
  
    public double berat;  
    public String makanan;  
    public String habitat;  
  
    public Hewan() {  
  
    }  
  
    public Hewan(double berat, String makanan, String habitat) {  
        this.berat = berat;  
        this.makanan = makanan;  
        this.habitat = habitat;  
    }  
  
    public abstract void bergerak();  
    public abstract void bernapas();  
  
    public void cetakInfo() {  
        System.out.println("Berat      : " + this.berat);  
        System.out.println("Makanan   : " + this.makanan);  
        System.out.println("Habitat  : " + this.habitat);  
    }  
}
```

2) Akan muncul error sebagai berikut

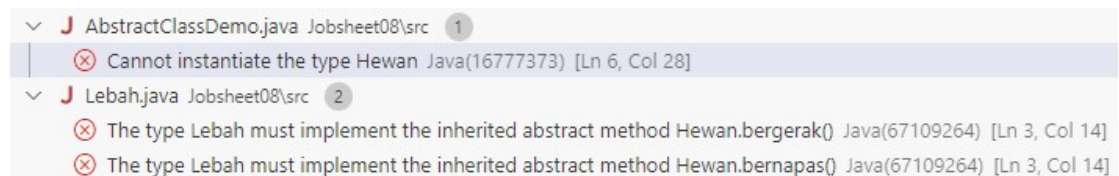
```
The type Hewan must be an abstract class to define abstract methods Java(16777549)  
Jobsheet08.src.Hewan  
View Problem (Alt+F8) Quick Fix... (Ctrl+.)
```

- 3) Ubah class Hewan menjadi abstract Class. Jalankan program kemudian amati hasilnya.

```
public abstract class Hewan {  
  
    public double berat;  
    public String makanan;  
    public String habitat;  
  
    public Hewan() {  
  
    }  
  
    public Hewan(double berat, String makanan, String habitat) {  
        this.berat = berat;  
        this.makanan = makanan;  
        this.habitat = habitat;  
    }  
  
    public abstract void bergerak();  
    public abstract void bernapas();  
  
    public void cetakInfo() {  
        System.out.println("Berat      : " + this.berat);  
        System.out.println("Makanan   : " + this.makanan);  
        System.out.println("Habitat  : " + this.habitat);  
    }  
}
```

- 4) Ubah class demo sebagai berikut. Run program kemudian amati hasilnya

```
public class AbstractClassDemo {  
    Run | Debug  
    public static void main(String[] args) {  
  
        Hewan hewan1 = new Hewan(10, "Rumput", "Savana");  
        hewan1.cetakInfo();  
        hewan1.bergerak();  
        hewan1.bernapas();  
    }  
}
```



The screenshot shows the bottom of an IDE window with two error messages. The first message is for 'AbstractClassDemo.java' at line 6, column 28, stating 'Cannot instantiate the type Hewan'. The second message is for 'Lebah.java' at line 3, column 14, stating 'The type Lebah must implement the inherited abstract method Hewan.bergerak()' and 'The type Lebah must implement the inherited abstract method Hewan.bernapas()'. Both messages are preceded by a red 'X' icon and a Java logo.

```
AbstractClassDemo.java Jobsheet08\src 1  
✗ Cannot instantiate the type Hewan Java(16777373) [Ln 6, Col 28]  
Lebah.java Jobsheet08\src 2  
✗ The type Lebah must implement the inherited abstract method Hewan.bergerak() Java(67109264) [Ln 3, Col 14]  
✗ The type Lebah must implement the inherited abstract method Hewan.bernapas() Java(67109264) [Ln 3, Col 14]
```

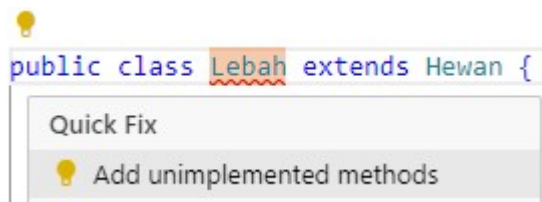
- 5) Ubah class demo sebagai berikut. Run program kemudian amati hasilnya

```
public class AbstractClassDemo {
    Run | Debug
    public static void main(String[] args) {

        Lebah lebah1 = new Lebah(kasta:"Ratu", berat:0.05, makanan:"Nektar", habitat:"Hutan");
        lebah1.cetakInfo();
        lebah1.bergerak();
        lebah1.bernapas();
    }
}
```

Berat : 0.05  
Makanan : Nektar  
Habitat : Hutan  
Exception in thread "main" java.lang.Error: Unresolved compilation problem:  
The type Lebah must implement the inherited abstract method Hewan.bergerak()  
at Jobsheet08.src.Lebah.bergerak(Lebah.java:3)  
at Jobsheet08.src.AbstractClassDemo.main(AbstractClassDemo.java:8)

- 6) Klik icon lampu pada class Lebah, kemudian pilih option "Implement all abstract method"



- 7) Implementasi method bergerak dan bernapas pada class Lebah sebagai berikut. Run program kemudian amati hasilnya.

```
public void bernapas() {
    System.out.println(x:"Otot perut mengendut, udara masuk melalui lubang di segmen tubuh");
    System.out.println(x:"Trakea pengiriman oksigen");
    System.out.println(x:"Otot perut berkontraksi, udara dikeluarkan");
}

public void bergerak() {
    System.out.println(x:"Mengepakkan sayap ke depan");
    System.out.println(x:"Memutar sayap hampir 90 derajat");
    System.out.println(x:"Mengepakkan sayap ke belakang");
}
```



- 8) Tambahkan method cetakInfo() pada class Lebah. Run program kemudian amati hasilnya.

```
@Override
public void cetakInfo() {
    super.cetakInfo();
    System.out.println("Kasta      : " + this.kasta);
}
```

```
Berat      : 0.05
Makanan    : Nektar
Habitat    : Hutan
Kasta      : Ratu
Mengepakkan sayap ke depan
Memutar sayap hampir 90 derajat
Mengepakkan sayap ke belakang
Otot perut mengendut, udara masuk melalui lubang di segmen tubuh
Trakea mengirimkan oksigen
Otot perut berkontraksi, udara dikeluarkan
```

- 9) Buat class Ular kemudian sebagai berikut. Instansiasi objek bertipe Ular pada class AbstractClassDemo. Eksekusi ketiga method untuk object tersebut.

```
public class Ular extends Hewan {

    public boolean isBerbisa;

    public Ular(boolean isBerbisa, double berat, String makanan, String habitat) {
        super(berat, makanan, habitat);
        this.isBerbisa = isBerbisa;
    }

    @Override
    public void bergerak() {
        System.out.println(x:"Mengerutkan otot dari segala sisi hingga membentuk lengkungan");
        System.out.println(x:"Menemukan titik penahan seperti batu atau pohon");
        System.out.println(x:"Menggunakan sisik untuk mendorong tubuh ke depan");
    }

    @Override
    public void bernapas() {
        System.out.println(x:"Otot tulang rusuk kontraksi, udara masuk lewat hidung");
        System.out.println(x:"trakea mengirimkan udara ke paru-paru");
        System.out.println(x:"Otot tulang rusuk relaksasi, udara dikeluarkan lewat hidung");
    }

    @Override
    public void cetakInfo() {
        super.cetakInfo();
        System.out.println("Berbisa      : " + (this.isBerbisa ? "Ya" : "Tidak"));
    }
}
```

## D. Pertanyaan Percobaan 2 (Pewarisan)

- 1) Pada langkah 1, mengapa sebaiknya method bergerak() dan bernapas() dideklarasikan sebagai abstract method?

Karena setiap hewan dapat bergerak dan bernapas namun berbeda-beda cara bernapasnya.

- 2) Mengapa pada langkah 2 muncul error?

Karena setiap method abstract maka class nya juga harus abstract.

- 3) Apakah sebuah class yang memiliki abstract method harus dideklarasikan sebagai abstract class?

Iya, karena menunjukkan bahwa class tersebut tidak lengkap dan tidak bisa di instansiasi secara langsung.

- 4) Sebaliknya, apakah abstract class harus memiliki abstract method?

Abstract class tidak harus memiliki abstract method karena bisa saja kebutuhannya hanya agar sebuah class tersebut tidak dijadikan sebuah objek.

- 5) Mengapa muncul error pada langkah 4?

Karena ketika sebuah class dideklarasikan sebagai abstract maka class tersebut tidak dapat dijadikan sebagai objek.

- 6) Apakah abstract class dapat memiliki constructor?

Abstract class dapat memiliki constructor layaknya class pada umumnya.

- 7) Apakah constructor abstract class dapat dipanggil?

Constructor dapat dipanggil namun tidak secara langsung melainkan dengan super.

- 8) Pada langkah 6-8, mengapa method bergerak() dan bernapas() harus di-override, namun method cetakInfo() tidak harus di-override?

Karena method bergerak() dan bernapas() masih kosong pada class Hewan, maka harus dioverride pada class child nya atau Lebah dan Ular.

- 9) Simpulkan kegunaan dari abstract method

- Ketika memiliki method dengan behavior yang sama namun cara atau implementasinya berbeda

- 10) Simpulkan kegunaan dari abstract class

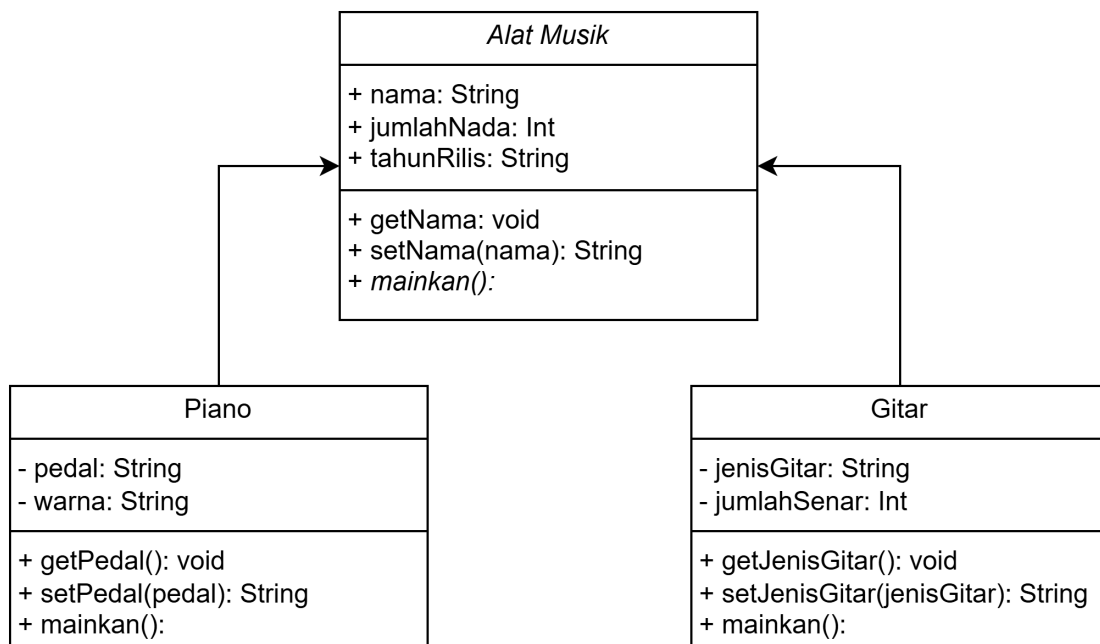
- Agar sebuah class tidak dapat dijadikan sebagai objek
- Sebagai panduan untuk subclass



## E. Tugas

Implementasikan class diagram yang telah dirancang pada tugas PBO Teori ke dalam ode program. Selanjutnya buatlah instansiasi objek dari masing-masing subclass emudian coba eksekusi method-method yang dimiliki.

### YAMAHA



## F. Jawaban Tugas

### 1) Program pada class AlatMusik

```
public abstract class AlatMusik {  
  
    public String nama;  
    public int jumlahNada;  
    public String tahunRilis;  
  
    public AlatMusik() {  
  
    }  
  
    public AlatMusik(String nama, int jumlahNada, String tahunRilis) {  
        this.nama = nama;  
        this.jumlahNada = jumlahNada;  
        this.tahunRilis = tahunRilis;  
    }  
  
    public String getNama() {  
        return nama;  
    }  
  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
  
    public int getJumlahNada() {  
        return jumlahNada;  
    }  
  
    public void setJumlahNada(int jumlahNada) {  
        this.jumlahNada = jumlahNada;  
    }  
  
    public String getTahunRilis() {  
        return tahunRilis;  
    }  
  
    public void setTahunRilis(String tahunRilis) {  
        this.tahunRilis = tahunRilis;  
    }  
  
    public void displayInfo() {  
        System.out.println("Nama      : " + nama);  
        System.out.println("Jumlah Nada : " + jumlahNada);  
        System.out.println("Tahun Rilis : " + tahunRilis);  
    }  
  
    public abstract void mainkan();  
  
}
```

## 2) Program pada class Piano

```
public class Piano extends AlatMusik {  
    private String pedal;  
    private String warna;  
  
    public Piano() {  
    }  
  
    public Piano(String nama, int jumlahNada, String tahunRilis, String pedal, String warna) {  
        super(nama, jumlahNada, tahunRilis);  
        this.pedal = pedal;  
        this.warna = warna;  
    }  
  
    public String getPedal() {  
        return pedal;  
    }  
  
    public void setPedal(String pedal) {  
        this.pedal = pedal;  
    }  
  
    public String getWarna() {  
        return warna;  
    }  
  
    public void setWarna(String warna) {  
        this.warna = warna;  
    }  
  
    public void displayInfo() {  
        super.displayInfo();  
        System.out.println("Pedal      : " + pedal);  
        System.out.println("Warna      : " + warna);  
    }  
  
    public void mainkan() {  
        System.out.println(x:"Dimainkan dengan menekan keyboard dan pedal");  
    }  
}
```

### 3) Program pada class Gitar

```
public class Gitar extends AlatMusik {  
  
    private String jenisGitar;  
    private int jumlahSenar;  
  
    public Gitar() {  
  
    }  
  
    public Gitar(String nama, int jumlahNada, String tahunRilis, String jenisGitar, int jumlahSenar) {  
        super(nama, jumlahNada, tahunRilis);  
        this.jenisGitar = jenisGitar;  
        this.jumlahSenar = jumlahSenar;  
    }  
  
    public String getJenisGitar() {  
        return jenisGitar;  
    }  
  
    public void setJenisGitar(String jenisGitar) {  
        this.jenisGitar = jenisGitar;  
    }  
  
    public int getJumlahSenar() {  
        return jumlahSenar;  
    }  
  
    public void setJumlahSenar(int jumlahSenar) {  
        this.jumlahSenar = jumlahSenar;  
    }  
  
    public void mainkan() {  
        System.out.println(x:"Dimainkan dengan dipetik");  
    }  
  
    public void displayInfo() {  
        super.displayInfo();  
        System.out.println("Jenis Gitar : " + jenisGitar);  
        System.out.println("Jumlah Senar : " + jumlahSenar);  
    }  
}
```

#### 4) Program pada class DemoAlatMusik

```
public class DemoAlatMusik {  
    Run | Debug  
    public static void main(String[] args) {  
  
        header();  
        System.out.println(x:"          Alat Musik Piano          ");  
        header();  
  
        Piano piano1 = new Piano(nama:"Yamaha P-525", jumlahNada:88, tahunRilis:"2023", pedal:"Ada", warna:"Hitam");  
        piano1.displayInfo();  
        piano1.mainkan();  
        header();  
  
        header();  
        System.out.println(x:"          Alat Musik Gitar          ");  
        header();  
  
        Gitar gitar1 = new Gitar(nama:"Yamaha LL16D-ARE", jumlahNada:12, tahunRilis:"2014", jenisGitar:"Akustik", jumlah:12);  
        gitar1.displayInfo();  
        gitar1.mainkan();  
        header();  
    }  
  
    public static void header() {  
        int length = 40;  
        for (int i = 0; i < length; i++) {  
            System.out.print(s:"=");  
        }  
        System.out.println(x:"");  
    }  
}
```

#### 5) Output

```
=====
                        Alat Musik Piano
=====
Nama       : Yamaha P-525
Jumlah Nada : 88
Tahun Rilis : 2023
Pedal      : Ada
Warna      : Hitam
Dimainkan dengan menekan keyboard dan pedal
=====
=====
                        Alat Musik Gitar
=====
Nama       : Yamaha LL16D-ARE
Jumlah Nada : 12
Tahun Rilis : 2014
Jenis Gitar : Akustik
Jumlah Senar : 12
Dimainkan dengan dipetik
=====
```