# LAPORAN TUGAS

# PEMROGRAMAN BERBASIS OBJEK

# PERTEMUAN KEDUA 29 AGUSTUS 2023

UIN SUNAN AMPEL
S U R A B A Y A

## DOSEN PEMBIMBING

Bayu Adhi Nugroho, Ph.D.

(197905182014031001)

## DISUSUN OLEH

Mochamad Roiyan Rintiarno

(09020622033)

## UIN SUNAN AMPEL SURABAYA

## TAHUN 2023

1. **Tugas**

→ Membuat kelas MakhlukHidup dan turunannya, dengan contoh satu makhluk hidup

→ Menggunakan konstruktor minimal 3 pada setiap kelas

→ Menggunakan keyword super dan this

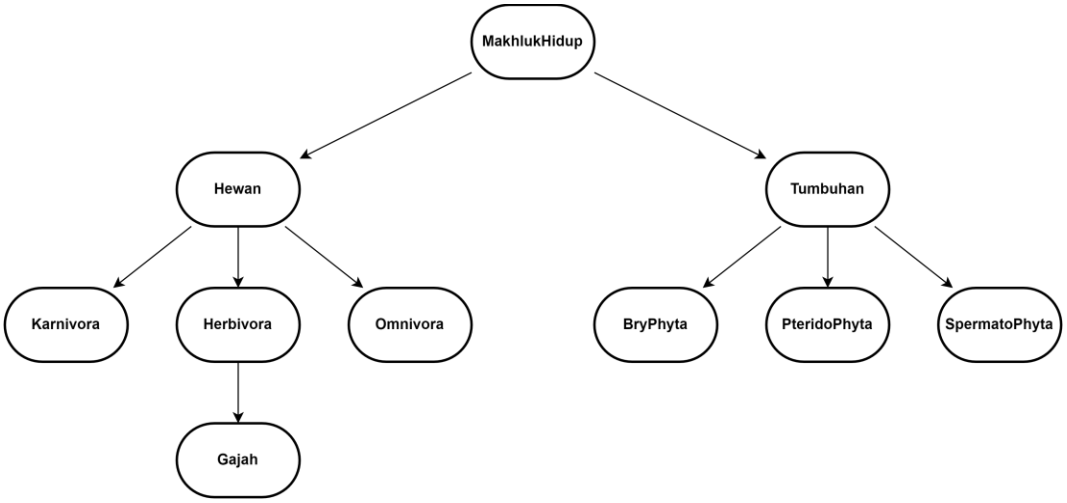
→ Membuat diagram kelas

→ Minimal kelas adalah 5

2. **Pembahasan dan Isi**

a. Persiapan dan Langkah-langkah

1. Membuat Diagram




Dari diagram tersebut dapat diketahui jika parent class yang akan dibuat adalah MakhlukHidup, lalu kemudian kedua class tersebut dikelompokkan atau dibagi kembali dengan menambahkan Hewan dan Tumbuhan. Kemudian pada hewan akan dikelompokkan sesuai dengan makanannya, yakni karnivora (pemakan daging) herbivora (pemakan tumbuhan) dan omnivora (pemakan segala). Lalu pada tumbuhan juga dibagi kembali menjadi tiga, yakni BryPhyta (tumbuhan lumut), PteridoPhyta (tumbuhan paku), dan SpermatoPhyta (tumbuhan berbiji). Dan yang terakhir, karena output yang diinginkan adalah gajah, maka kelas Gajah akan diletakkan dibawah kelas Herbivora. Lalu selanjutnya kelas tersebut akan dipanggil melalui kelas lain diluar package MakhlukHidup tersebut agar pembaca sekalian dapat mengetahui perbedaan public, private, dan protected.

2. Membuat Kelas Parent dengan nama MakhlukHidup

```java
package MakhlukHidup;

import java.util.ArrayList;

public class MakhlukHidup {
    private String nama;
    private String asal;
    private String namaFamily;
    protected String jenisMakanan;
    protected ArrayList<String> ciriCiri = new ArrayList<String>();

    public MakhlukHidup() {
        System.out.println("⇒Ini adalah konstruktor kelas MakhlukHidup tanpa parameter");
    }

    public MakhlukHidup(String nama) {
        System.out.println("⇒Ini adalah konstruktor kelas MakhlukHidup dengan parameter nama, yakni " + nama);
        this.nama = nama;
    }

    public MakhlukHidup(String nama, String asal) {
        System.out.println(
                "⇒Ini adalah konstruktor kelas MakhlukHidup dengan parameter nama, yakni " + nama
                        + " dan parameter asal, yakni " + asal);
        this.nama = nama;
        this.asal = asal;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getAsal() {
        return asal;
    }

    public void setAsal(String asal) {
        this.asal = asal;
    }

    public String getNamaFamily() {
        return namaFamily;
    }

    public void setNamaFamily(String namaFamily) {
        this.namaFamily = namaFamily;
    }
}
```

Kelas MakhlukHidup adalah parent dari kelas-kelas setelahnya, terdapat beberapa atribut, yakni nama, asal, dan namaFamily yang bersifat private (hanya dapat diakses didalam kelas saja), sehingga diperlukan method untuk mengaksesnya secara publik, hal ini dinamakan dengan getter dan setter. Dimana atribut nama memiliki method getter yakni getNama dan setter yakni setNama, lalu kemudian atribut asal yang memiliki method getter getAsal dan setter yakni

setAsal, lalu yang terakhir terdapat atribut namaFamily yang memiliki method getter getNamaFamily dan setter yakni setNamaFamily. Lalu kemudian ada atribut jenisMakanan dan ciriCiri yang bersifat protected, yakni atribut tersebut hanya dapat diakses didalam kelas itu sendiri dan didalam kelas turunannya. Pada kelas ini tidak dibuatkan setter dan getternya.

Pada kelas ini juga terdapat beberapa konstruktor, yakni yang pertama tanpa parameter, dimana jika kelas MakhlukHidup tersebut dipanggil, maka akan mengeluarkan output "=>Ini adalah konstruktor kelas MakhlukHidup tanpa parameter". Lalu pada konstruktor kedua terdapat satu buah parameter, yakni nama yang bertipe data String, dimana jika kelas MakhlukHidup dipanggil, maka akan keluar output "=>Ini adalah konstruktor kelas MakhlukHidup dengan parameter nama, yakni " + nama. Variabel nama berasal dari inputan yang dimasukkan oleh user. Lalu pada konstruktor ketiga, terdapat dua buah parameter, yakni nama (String) dan asal (String), yang memiliki output "=>Ini adalah konstruktor kelas MakhlukHidup dengan parameter nama, yakni " + nama + " dan parameter asal, yakni " + asal.

3. Membuat Kelas Child pertama, yakni kelas Hewan dan kelas Tumbuhan
   a. Kelas Hewan

```java
package MakhlukHidup;

import java.util.ArrayList;

public class Hewan extends MakhlukHidup {
    public int jumlahKaki = 0;

    public Hewan() {
        super("gajah");
        System.out.println("=>=>Ini adalah konstruktor kelas Hewan tanpa parameter");
    }

    public Hewan(String nama) {
        super(nama);
        System.out.println("=>=>Ini adalah konstruktor kelas Hewan dengan parameter nama, yakni " + nama);
    }

    public Hewan(String nama, String asal) {
        super(nama, asal);
        System.out.println("=>=>Ini adalah konstruktor kelas Hewan dengan parameter nama, yakni " + nama
                + " dan parameter asal, yakni " + asal);
    }

    public ArrayList<String> getCiriCiri() {
        return ciriCiri;
    }

    public void addCiriCiri(String ciriCiri) {
        this.ciriCiri.add(ciriCiri);
    }
}

```

Pada gambar diatas, kelas Hewan mewarisi kelas MakhlukHidup, hal ini ditandai dengan adanya kata kunci extends yang berarti setiap metod atau atribut yang berada pada kelas Makhluk hidup yang bersifat public/protected akan dapat digunakan pada kelas Hewan, tanpa menduplikat kodenya. Kelas Hewan memiliki 1 atribut yakni jumlahKaki yang bersifat publik, sehingga atribut tersebut dapat diakses oleh seluruh yang memanggilnya. Lalu terdapat setter dan getter atribut ciriCiri yakni getCiriCiri dan addCiriCiri. Pada kelas Hewan, kita tidak perlu menambahkan atribut ciriCiri karena hal tersebut telah ditambahkan dalam kelas induknya (parent) dan bersifat protecteed sehingga dapat dipanggil di kelas turunannya.

Pada kelas Hewan juga terdapat 3 konstruktor yang masing-masing terdiri dari berbagai parameter, ada yang tidak menggunakan parameter, ada yang hanya 1, lalu ada juga yang mempunyai 2 parameter. Pada kontruktor yang pertama, output yang akan keluar adalah "=>=>Ini adalah konstruktor kelas Hewan tanpa parameter". Lalu pada konstruktor yang kedua terdapat satu buah parameter, yakni nama, yang outputnya menjadi “=>=>Ini adalah konstruktor kelas Hewan dengan

parameter nama, yakni " + nama. Kemudian pada konstruktor yang ketiga terdapat 2 buah parameter, yakni nama dan asal, yang mempunyai output =>=>Ini adalah konstruktor kelas Hewan dengan parameter nama, yakni " + nama + " dan parameter asal, yakni " + asal.

b. Kelas Tumbuhan

```java
package MakhlukHidup;

public class Tumbuhan extends MakhlukHidup {
    private String bentukDaun;
    protected String bentukBiji;

    public Tumbuhan() {
        System.out.println("Ini adalah konstruktor kelas Tumbuhan tanpa parameter");
    }

    public Tumbuhan(String bentukDaun) {
        System.out
                .println("Ini adalah konstruktor kelas Tumbuhan dengan satu parameter, yakni bentukDaun " + bentukDaun);
        this.bentukDaun = bentukDaun;
    }

    public Tumbuhan(String bentukDaun, String warnaDaun) {
        System.out.println("Ini adalah konstruktor kelas Tumbuhan dengan dua parameter, yakni bentukDaun " + bentukDaun
                + " dan warnaDaun " + warnaDaun);
    }

    public void setBentukDaun(String bentukDaun) {
        this.bentukDaun = bentukDaun;
    }

    public String getBentukDaun() {
        return bentukDaun;
    }
}

```

Pada gambar tersebut, kelas Tumbuhan akan mewarisi kelas MakhlukHidup sehingga semua method dan atribut yang berada pada kelas MakhlukHidup juga akan terdapat didalam kelas Tumbuhan.

4. Membuat Kelas Child untuk Kelas Hewan, yakni kelas Karnivora, Herbivora, dan Omnivora

   a. Kelas Karnivora

```java
package MakhlukHidup;

public class Karnivora extends Hewan {
    public Karnivora() {
        this.jenisMakanan = "Karnivora";
        System.out.println("==>==>Ini adalah konstruktor kelas Karnivora tanpa parameter");
    }

    public Karnivora(String jenisMakanan) {
        System.out.println(
                "==>==>Ini adalah konstruktor kelas Karnivora dengan satu parameter, yakni jenisMakanan: "
                        + jenisMakanan);
        this.jenisMakanan = jenisMakanan;
    }

    public Karnivora(String jenisMakanan, int jumlahKaki) {
        super();
        System.out.println(
                "==>==>Ini adalah konstruktor kelas Karnivora dengan parameter jenisMakanan, yakni " + jenisMakanan
                        + " dan jumlahKaki, yakni " + jumlahKaki);
        this.jenisMakanan = jenisMakanan;
        this.jumlahKaki = jumlahKaki;
    }

    public Karnivora(String namaHewan, String asal, int jumlahKaki) {
        System.out.println("==>==>Ini adalah hewan " + namaHewan + " yang berasal dari " + asal
                + " dan memiliki kaki berjumlah " + jumlahKaki);
    }

    public String getJenisMakanan() {
        return jenisMakanan;
    }
}

```

b. Kelas Herbivora

```java
package MakhlukHidup;

public class Herbivora extends Hewan {

    public Herbivora() {
        this.jenisMakanan = "Herbivora";
        System.out.println("=>=>=>Ini adalah konstruktor kelas Herbivora tanpa parameter");
    }

    public Herbivora(String jenisMakanan) {
        System.out.println(
                "=>=>=>Ini adalah konstruktor kelas Herbivora dengan satu parameter, yakni jenisMakanan: "
                        + jenisMakanan);
        this.jenisMakanan = jenisMakanan;
    }

    public Herbivora(String jenisMakanan, int jumlahKaki) {
        super();
        System.out.println(
                "=>=>=>Ini adalah konstruktor kelas Herbivora dengan parameter jenisMakanan, yakni " + jenisMakanan
                        + " dan jumlahKaki, yakni " + jumlahKaki);
        this.jenisMakanan = jenisMakanan;
        this.jumlahKaki = jumlahKaki;
    }

    public Herbivora(String namaHewan, String asal, int jumlahKaki) {
        System.out.println("=>=>=>Ini adalah hewan " + namaHewan + " yang berasal dari " + asal
                + " dan memiliki kaki berjumlah " + jumlahKaki);
    }

    public String getJenisMakanan() {
        return jenisMakanan;
    }
}

```

Pada gambar tersebut, kelas Herbivora akan mewarisi kelas Hewan sehingga semua method dan atribut yang berada pada kelas Hewan juga akan terdapat didalam kelas Herbivora. Di dalam kelas Herbivora juga terdapat method constructor, yakni method yang akan langsung dipanggil saat kita memanggil kelas tersebut, sebagaimana contoh pada kelas Herbivora diatas, yakni ketika kelas tersebut dipanggil, maka akan langsung meng-set atribut jenisMakanan yang terdapat pada kelas parent (dalam hal ini method tersebut terdapat pada kelas MakhlukHidup), sehingga atribut jenisMakanan akan langsung memiliki value “Herbivora” yang bertipe data String. Konstruktor yang kedua mempunyai satu buah parameter, yakni jenisMakanan, lalu pada konstruktor yang ketiga terdapat dua buah parameter, yakni jenisMakanan dan jumlahKaki. Serta pada kontruktor keempat terdapat 3 buah parameter, yakni namaHewan, asal, dan jumlahKaki. Lalu didalam kelas tersebut juga terdapat method getter yang bernama getJenisMakanan.

c. Kelas Omnivora

```java
package MakhlukHidup;

public class Omnivora extends Hewan {
    public Omnivora() {
        this.jenisMakanan = "Omnivora";
        System.out.println("==>==>Ini adalah konstruktor kelas Omnivora tanpa parameter");
    }

    public Omnivora(String jenisMakanan) {
        System.out.println(
                "==>==>Ini adalah konstruktor kelas Omnivora dengan satu parameter, yakni jenisMakanan:  "
                        + jenisMakanan);
        this.jenisMakanan = jenisMakanan;
    }

    public Omnivora(String jenisMakanan, int jumlahKaki) {
        super();
        System.out.println(
                "==>==>Ini adalah konstruktor kelas Omnivora dengan parameter jenisMakanan, yakni " + jenisMakanan
                        + " dan jumlahKaki, yakni " + jumlahKaki);
        this.jenisMakanan = jenisMakanan;
        this.jumlahKaki = jumlahKaki;
    }

    public Omnivora(String namaHewan, String asal, int jumlahKaki) {
        System.out.println("==>==>Ini adalah hewan " + namaHewan + " yang berasal dari " + asal
                + " dan memiliki kaki berjumlah " + jumlahKaki);
    }

    public String getJenisMakanan() {
        return jenisMakanan;
    }
}

```

5. Membuat Kelas Child untuk kelas Tumbuhan, yakni kelas BryPhyta, PteridoPhyta, SpermatoPhyta
    a. Kelas BryPhyta

```java
public BryPhyta() {
        this.jenisMakanan = "Daun";
        System.out.println("==>==>Ini adalah konstruktor kelas BryPhyta tanpa parameter");
    }

    public BryPhyta(String jenisMakanan) {
        System.out.println(
                "==>==>Ini adalah konstruktor kelas BryPhyta dengan satu parameter, yakni jenisMakanan:  "
                        + jenisMakanan);
        this.jenisMakanan = jenisMakanan;
    }

    public BryPhyta(String jenisMakanan, String bentukDaun) {
        super();
        System.out.println(
                "==>==>Ini adalah konstruktor kelas BryPhyta dengan parameter jenisMakanan, yakni " + jenisMakanan
                        + " dan bentukDaun, yakni " + bentukDaun);
        this.jenisMakanan = jenisMakanan;
    }
```

    b. Kelas PteridoPhyta

```java
package MakhlukHidup;

public class PteridoPhyta extends Tumbuhan {
    public PteridoPhyta() {
        this.jenisMakanan = "Daun";
        System.out.println("=>=>=>Ini adalah konstruktor kelas PteridoPhyta tanpa parameter");
    }

    public PteridoPhyta(String jenisMakanan) {
        System.out.println(
                "=>=>=>Ini adalah konstruktor kelas PteridoPhyta dengan satu parameter, yakni jenisMakanan:  "
                        + jenisMakanan);
        this.jenisMakanan = jenisMakanan;
    }

    public PteridoPhyta(String jenisMakanan, String bentukDaun) {
        super();
        System.out.println(
                "=>=>=>Ini adalah konstruktor kelas PteridoPhyta dengan parameter jenisMakanan, yakni " + jenisMakanan
                        + " dan bentukDaun, yakni " + bentukDaun);
        this.jenisMakanan = jenisMakanan;
    }
}

```

c. Kelas SpermatoPhyta

```java
package MakhlukHidup;

public class SpermatoPhyta extends Tumbuhan {
    public SpermatoPhyta() {
        this.jenisMakanan = "Daun";
        System.out.println("=>=>=>Ini adalah konstruktor kelas SpermatoPhyta tanpa parameter");
    }

    public SpermatoPhyta(String jenisMakanan) {
        System.out.println(
                "=>=>=>Ini adalah konstruktor kelas SpermatoPhyta dengan satu parameter, yakni jenisMakanan:  "
                        + jenisMakanan);
        this.jenisMakanan = jenisMakanan;
    }

    public SpermatoPhyta(String jenisMakanan, String bentukDaun) {
        super();
        System.out.println(
                "=>=>=>Ini adalah konstruktor kelas SpermatoPhyta dengan parameter jenisMakanan, yakni " + jenisMakanan
                        + " dan bentukDaun, yakni " + bentukDaun);
        this.jenisMakanan = jenisMakanan;
    }
}

```

6. Membuat package baru bernama Output dengan nama kelas CobaOutput untuk mencoba package MakhlukHidup sebelumnya, menggunakan package baru bertujuan agar dapat melihat hasil implementasi dari parameter protected

```java
package Output;

import MakhlukHidup.Herbivora;

public class OutputTugas {
    public static void main(String[] args) {
        Herbivora gajah;
        gajah = new Herbivora();
        gajah = new Herbivora("daun jati");
        gajah = new Herbivora("empal", 4);
    }
}
```

Pada kelas Output tersebut, walaupun hanya meng-instance objek yang berada di dalam kelas Herbivora, instance gajah akan langsung menjalankan method konstruktor dikarenakan konstruktor adalah method yang berjalan diawal saat kelas diinisialisasi.

7. Hasil/Output

```
milea@khaylila:~/Documents/kuliah/pemrogramanBerbasisObyek/TugasPertemuan2/build/classes$ java Output.OutputTugas
=>Ini adalah konstruktor kelas MakhlukHidup dengan parameter nama, yakni gajah
=>=>Ini adalah konstruktor kelas Hewan tanpa parameter
=>=>=>Ini adalah konstruktor kelas Herbivora tanpa parameter
=>Ini adalah konstruktor kelas MakhlukHidup dengan parameter nama, yakni gajah
=>=>Ini adalah konstruktor kelas Hewan tanpa parameter
=>=>=>Ini adalah konstruktor kelas Herbivora dengan satu parameter, yakni jenisMakanan: daun jati
=>Ini adalah konstruktor kelas MakhlukHidup dengan parameter nama, yakni gajah
=>=>Ini adalah konstruktor kelas Hewan tanpa parameter
=>=>=>Ini adalah konstruktor kelas Herbivora dengan parameter jenisMakanan, yakni empal dan jumlahKaki, yakni 4
milea@khaylila:~/Documents/kuliah/pemrogramanBerbasisObyek/TugasPertemuan2/build/classes$
```

Pada percobaan diatas, output yang dihasilkan berbeda-beda, sesuai dengan konstruktor yang dipanggil. Pada konstruktor pertama tidak memanggil apapun, sehingga output yang keluar adalah ”Ini adalah konstruktor kelas Herbivora tanpa parameter”, lalu yang kedua ditambahkan variabel String pada parameter pertama, sehingga output yang keluar adalah ”Ini adalah konstruktor kelas Herbivora dengan satu parameter, yakni jenisMakanan: daun jati”, dan yang terakhir ditambahkannya dua parameter yakni, jenisMakanan yang bertipe String, dan jumlahKaki yang

bertipe integer, sehingga output yang keluar adalah ”Ini adalah konstruktor kelas Herbivora dengan parameter jenisMakanan, yakni empal dan jumlahKaki, yakni 4”.