

Nama : Arditya Adjie Rosandi
NIM : 20230801274
Pemrograman Berorientasi Objek



Bagian 1: Bangun Datar

=> Parent Class

```
public class BangunDatar{  
    float luas(){  
        System.out.print("\033[H\033[2J");  
        System.out.println("MENGHITUNG LUAS BANGUN DATAR.");  
        return 0;  
    }  
  
    float keliling(){  
        System.out.println("MENGHITUNG KELILING BANGUN DATAR");  
        System.out.println();  
        return 0;  
    }  
}
```

Kelas ini merupakan kelas dasar (parent class) untuk berbagai bangun datar. Ia menyediakan metode umum untuk menghitung luas dan keliling, yang dapat di-override oleh kelas turunan.

Metode luas():

- Menampilkan pesan "MENGHITUNG LUAS BANGUN DATAR" ke konsol.
- Membersihkan layar sebelum mencetak pesan menggunakan escape sequence `\033[H\033[2J`.
- Mengembalikan nilai default 0 sebagai placeholder.

Metode keliling():

- Menampilkan pesan "MENGHITUNG KELILING BANGUN DATAR" ke konsol.
- Mengembalikan nilai default 0 sebagai placeholder.

Kelas ini dirancang sebagai template, dengan logika perhitungan spesifik yang akan diimplementasikan di kelas turunan.

=> Child Class

PersegiPanjang (turunan dari BangunDatar).

```
class Persegi extends BangunDatar {
    float sisi;

    @Override
    float luas() {
        // Menghitung luas persegi dan mengembalikan hasilnya sebagai float
        float hasil = sisi * sisi;
        System.out.println("LUAS PERSEGI: " + hasil);
        return hasil;
    }

    @Override
    float keliling() {
        // Menghitung keli float sisi dan mengembalikan hasilnya sebagai float
        float hasil = 4 * sisi;
        System.out.println("KELILING PERSEGI : " + hasil);
        System.out.println();
        return hasil;
    }
}
```

Kelas ini merupakan turunan (**child class**) dari BangunDatar. Ia meng-override metode luas dan keliling untuk menghitung luas dan keliling persegi berdasarkan nilai properti sisi.

Properti sisi:

- Menyimpan panjang sisi persegi.

Override luas():

- Menghitung luas menggunakan rumus sisi * sisi.
- Menampilkan hasil perhitungan.

Override keliling():

- Menghitung keliling menggunakan rumus 4 * sisi.
- Menampilkan hasil perhitungan.

PersegiPanjang (turunan dari BangunDatar).

```
class PersegiPanjang extends BangunDatar {
    float panjang, lebar;

    @Override
    float luas() {
        // Menghitung luas persegi panjang dan mengembalikan hasilnya sebagai float
        float hasil = panjang * lebar;
        System.out.println("LUAS PERSEGI PANJANG : " + hasil);
        return hasil;
    }

    @Override
    float keliling() {
        // Menghitung keliling persegi panjang dan mengembalikan hasilnya sebagai float
        float hasil = 2 * (panjang + lebar);
        System.out.println("KELILING PERSEGI PANJANG : " + hasil);
        System.out.println();
        return hasil;
    }
}
```

Kelas ini merupakan turunan (child class) dari BangunDatar. Ia meng-override metode luas dan keliling untuk menghitung luas dan keliling persegi panjang berdasarkan nilai properti panjang dan lebar.

Properti panjang dan lebar:

- Menyimpan panjang dan lebar persegi panjang.

Override luas():

- Menghitung luas menggunakan rumus panjang * lebar.
- Menampilkan hasil perhitungan.

Override keliling():

- Menghitung keliling menggunakan rumus 2 * (panjang + lebar).
- Menampilkan hasil perhitungan.

Segitiga (turunan dari BangunDatar).

```
class Segitiga extends BangunDatar {
    float alas, tinggi;

    @Override
    float luas() {
        // Menghitung luas segitiga dan mengembalikan hasilnya sebagai float
        float hasil = 0.5f * alas * tinggi;
        System.out.println("LUAS SEGITIGA: " + hasil);
        return hasil;
    }

    @Override
    float keliling() {
        // Keliling segitiga belum didefinisikan dengan lengkap
        System.out.println("KELILING SEGITIGA: BELUM DIDEFINISIKAN (BUTUH SISI-SISI LAIN");
        System.out.println();
        return 0;
    }
}
```

Kelas ini merupakan turunan (child class) dari BangunDatar. Ia meng-override metode luas dan keliling untuk menghitung luas segitiga dan menampilkan pesan terkait keliling segitiga.

Properti alas dan tinggi:

- Menyimpan panjang alas dan tinggi segitiga.

Override luas():

- Menghitung luas menggunakan rumus $0.5 * \text{alas} * \text{tinggi}$.
- Menampilkan hasil perhitungan.

Override keliling():

- Menampilkan pesan bahwa keliling segitiga belum didefinisikan karena membutuhkan informasi sisi-sisi lain.
- Mengembalikan nilai 0 sebagai placeholder.

Lingkaran (turunan dari BangunDatar).

```
class Lingkaran extends BangunDatar {  
    float r;  
    final float PI = 3.14f;  
  
    @Override  
    float luas() {  
        // Menghitung luas lingkaran dan mengembalikan hasilnya sebagai float  
        float hasil = PI * r * r;  
        System.out.println("LUAS LINGKARAN : " + hasil);  
        return hasil;  
    }  
  
    @Override  
    float keliling() {  
        // Menghitung keliling lingkaran dan mengembalikan hasilnya sebagai float  
        float hasil = 2 * PI * r;  
        System.out.println("KELILING LINGKARAN : " + hasil);  
        System.out.println();  
        return hasil;  
    }  
}
```

Kelas ini merupakan turunan (child class) dari BangunDatar. Ia meng-override metode luas dan keliling untuk menghitung luas dan keliling lingkaran berdasarkan jari-jari.

Properti r (jari-jari):

- Menyimpan panjang jari-jari lingkaran.
- PI adalah konstanta dengan nilai 3.14.

Override luas():

- Menghitung luas menggunakan rumus $PI * r * r$.
- Menampilkan hasil perhitungan.

Override keliling():

- Menghitung keliling menggunakan rumus $2 * PI * r$.
- Menampilkan hasil perhitungan.

=> Main Class

```
public class Main {  
    Run | Debug  
    public static void main(String[] args) {  
        // Membuat objek Bangun Datar  
        BangunDatar bangunDatar = new BangunDatar();  
  
        // Membuat objek Persegi dan mengisi nilai properti  
        Persegi persegi = new Persegi();  
        persegi.sisi = 2;  
  
        // Membuat objek Lingkaran dan mengisi nilai properti  
        Lingkaran lingkaran = new Lingkaran();  
        lingkaran.r = 22;  
  
        // Membuat objek Persegi Panjang dan mengisi nilai properti  
        PersegiPanjang persegiPanjang = new PersegiPanjang();  
        persegiPanjang.panjang = 8;  
        persegiPanjang.lebar = 4;  
  
        // Membuat objek Segitiga dan mengisi nilai properti  
        Segitiga mSegitiga = new Segitiga();  
        mSegitiga.alas = 12;  
        mSegitiga.tinggi = 8;  
  
        // Memanggil method Luas dan Keliling  
        bangunDatar.luas();  
        bangunDatar.keliling();  
  
        persegi.luas();  
        persegi.keliling();  
  
        lingkaran.luas();  
        lingkaran.keliling();  
  
        persegiPanjang.luas();  
        persegiPanjang.keliling();  
  
        mSegitiga.luas();  
        mSegitiga.keliling();  
    }  
}
```

Kelas ini merupakan kelas utama untuk menguji fungsi dari berbagai objek bangun datar seperti persegi, lingkaran, persegi panjang, dan segitiga.

Pembuatan Objek:

- Membuat objek BangunDatar sebagai instance umum.
- Membuat objek Persegi dan mengisi nilai properti sisi.
- Membuat objek Lingkaran dan mengisi nilai properti r (jari-jari).
- Membuat objek PersegiPanjang dan mengisi nilai properti panjang dan lebar.
- Membuat objek Segitiga dan mengisi nilai properti alas dan tinggi.

Pemanggilan Metode:

- Memanggil metode luas() dan keliling() dari setiap objek untuk menghitung dan menampilkan hasil perhitungan masing-masing bangun datar.
- Metode di setiap objek akan menjalankan implementasi spesifik yang telah di-override dari kelas BangunDatar.

=> Output

```
MENGHITUNG LUAS BANGUN DATAR.  
MENGHITUNG KELILING BANGUN DATAR  
  
LUAS PERSEGI: 4.0  
KELILING PERSEGI : 8.0  
  
LUAS LINGKARAN : 1519.76  
KELILING LINGKARAN : 138.16  
  
LUAS PERSEGI PANJANG : 32.0  
KELILING PERSEGI PANJANG : 24.0  
  
LUAS SEGITIGA: 48.0  
KELILING SEGITIGA: BELUM DIDEFINISIKAN (BUTUH SISI-SISI LAIN
```

Bagian 2: Hewan

=> Parent Class

```
public class Hewan {  
    protected String suara;  
  
    public Hewan(){  
        this.suara = "SUARA";  
    }  
  
    public void cetak(){  
        System.err.println("HEWAN BERSUARA" +suara);  
    }  
}
```

Kelas ini merupakan kelas dasar (parent class) untuk objek hewan yang menyediakan properti dan metode umum.

Properti suara:

- Menyimpan suara hewan, dengan tipe data String.
- Properti ini bersifat protected, yang memungkinkan akses dari kelas ini dan kelas turunan.

Konstruktor Hewan():

- Menginisialisasi properti suara dengan nilai default "SUARA". Ini berarti setiap objek Hewan yang dibuat akan memiliki suara dengan nilai "SUARA", kecuali kelas turunan memberikan nilai yang berbeda.

Metode cetak():

- Menampilkan pesan "HEWAN BERSUARA" diikuti dengan nilai dari properti suara.
- Pesan dicetak menggunakan System.err.println(), yang menunjukkan output ke stream error.

```
class Mamalia extends Hewan {  
    protected String berkembangbiak;  
  
    public Mamalia(){  
        this.berkembangbiak = "BERANAK";  
    }  
  
    public void cetak(){  
        System.out.println("MAMALIA BERKEMBANG BIAK" +berkembangbiak);  
    }  
}
```

Kelas ini merupakan turunan (child class) dari Hewan yang menambahkan properti dan metode terkait mamalia.

Properti berkembangbiak:

- Menyimpan cara berkembang biak mamalia, dengan tipe data String.
- Properti ini diinisialisasi dengan nilai default "BERANAK", yang berarti mamalia berkembang biak dengan melahirkan.

Konstruktor Mamalia():

- Menginisialisasi properti berkembangbiak dengan nilai "BERANAK", menunjukkan cara berkembang biak mamalia.

Override cetak():

- Meng-override metode cetak() dari kelas Hewan untuk menampilkan pesan yang lebih spesifik, yaitu "MAMALIA BERKEMBANG BIAK" diikuti dengan nilai properti berkembangbiak.
- Pesan ini dicetak menggunakan System.out.println().

```

class Sapi extends Mamalia {
    private String nama;
    private String umur;

    public Sapi(){
        suara = "MOOOOOHHH";
        berkembangbiak = "BERANAK";
        this.nama = "AVRI";
        this.umur = "1";
    }

    public void cetak(){
        System.out.println("DATA HEWAN MAMALIA SAPI");
        System.out.println();
        System.out.println("SUARA : " + suara);
        System.out.println("BERKEMBANG BIAK : " + berkembangbiak);
        System.out.println("NAMA : " + nama);
        System.out.println("UMUR : " + umur + "TAHUN");
    }
}

```

Kelas ini merupakan turunan (child class) dari Mamalia yang lebih spesifik menggambarkan objek Sapi.

Properti nama dan umur:

- nama menyimpan nama sapi, dengan tipe data String.
- umur menyimpan umur sapi, dengan tipe data String.

Konstruktor Sapi():

- Menginisialisasi properti suara dari kelas Hewan dengan nilai "MOOOOOHHH", yang menggambarkan suara sapi.
- Menginisialisasi properti berkembangbiak dari kelas Mamalia dengan nilai "BERANAK", yang menunjukkan bahwa sapi berkembang biak dengan cara melahirkan.
- Menginisialisasi properti nama dengan nilai "AVRI" dan umur dengan nilai "1", yang menunjukkan nama dan umur sapi tersebut.

Override cetak():

- Meng-override metode cetak() dari kelas Mamalia untuk menampilkan informasi yang lebih spesifik mengenai sapi.
- Informasi yang ditampilkan meliputi suara, cara berkembang biak, nama, dan umur sapi.
- Setiap informasi ditampilkan menggunakan System.out.println() dengan format yang jelas.

Secara keseluruhan, kelas Sapi memperluas kelas Mamalia dengan menambahkan atribut nama dan umur, serta mengubah implementasi metode cetak() untuk menampilkan data yang lebih spesifik mengenai sapi.


```

class Kambing extends Mamalia {
    private String nama;
    private String umur;

    public Kambing(){
        suara = "MBEEEEKKK";
        berkembangbiak = "BERANAK";
        this.nama = "TEGAR";
        this.umur = "2";
    }

    public void cetak(){
        System.out.println("DATA HEWAN MAMALIA KAMBING");
        System.out.println("SUARA : " + suara);
        System.out.println("BERKEMBANG BIAK : " + berkembangbiak);
        System.out.println("NAMA : " + nama);
        System.out.println("UMUR : " + umur + "TAHUN");
    }
}

```

Kelas ini merupakan turunan (child class) dari Mamalia yang lebih spesifik menggambarkan objek Kambing.

Properti nama dan umur:

- nama menyimpan nama kambing, dengan tipe data String.
- umur menyimpan umur kambing, dengan tipe data String.

Konstruktor Kambing():

- Menginisialisasi properti suara dari kelas Hewan dengan nilai "MBEEEEKKK", yang menggambarkan suara kambing.
- Menginisialisasi properti berkembangbiak dari kelas Mamalia dengan nilai "BERANAK", yang menunjukkan bahwa kambing berkembang biak dengan cara melahirkan.
- Menginisialisasi properti nama dengan nilai "TEGAR" dan umur dengan nilai "2", yang menunjukkan nama dan umur kambing tersebut.

Override cetak():

- Meng-override metode cetak() dari kelas Mamalia untuk menampilkan informasi yang lebih spesifik mengenai kambing.
- Informasi yang ditampilkan meliputi suara, cara berkembang biak, nama, dan umur kambing.
- Setiap informasi ditampilkan menggunakan System.out.println() dengan format yang jelas.

Secara keseluruhan, kelas Kambing memperluas kelas Mamalia dengan menambahkan atribut nama dan umur, serta mengubah implementasi metode cetak() untuk menampilkan data yang lebih spesifik mengenai kambing.

```
class Unggas extends Hewan {  
    protected String berkembangBiak;  
  
    public Unggas(){  
        this.berkembangBiak = "BERTELUR";  
    }  
  
    public void cetak(){  
        System.out.println("BERTELUR");  
    }  
}
```

Kelas ini merupakan turunan (child class) dari Hewan yang menggambarkan objek Unggas (seperti burung atau ayam).

Properti berkembangBiak:

- Menyimpan cara berkembang biak unggas, dengan tipe data String.
- Properti ini diinisialisasi dengan nilai default "BERTELUR", yang menunjukkan bahwa unggas berkembang biak dengan cara bertelur.

Konstruktur Unggas():

- Menginisialisasi properti berkembangBiak dengan nilai "BERTELUR", yang menunjukkan cara berkembang biak unggas.

Override cetak():

- Meng-override metode cetak() dari kelas Hewan untuk menampilkan informasi yang lebih spesifik mengenai unggas.
- Dalam metode ini, hanya mencetak informasi mengenai cara berkembang biak unggas, yaitu "BERTELUR", menggunakan System.out.println().

Secara keseluruhan, kelas Unggas memperluas kelas Hewan dengan menambahkan properti khusus yang menggambarkan cara berkembang biak unggas dan meng-override metode cetak() untuk menampilkan informasi ini.

```

class Burung extends Unggas {
    private String jenis;
    private String ciri;

    public Burung () {
        suara = "MBEEKURR";
        berkembangBiak = "BERTELUR";
        this.jenis = "MERPATI";
        this.ciri = "BULU PUTIH";
    }

    public void cetak(){
        System.out.println("DATA HEWAN UNGGAS BURUNG");
        System.out.println("SUARA : " + suara);
        System.out.println("BERKEMBANG BIAK : " + berkembangBiak);
        System.out.println("JENIS : " + jenis);
        System.out.println("CIRI : " + ciri);
    }
}

```

Kelas ini merupakan turunan (child class) dari Unggas yang menggambarkan objek Burung lebih spesifik.

Properti jenis dan ciri:

- jenis menyimpan jenis burung, dengan tipe data String.
- ciri menyimpan ciri khas burung, dengan tipe data String.

Konstruktor Burung():

- Menginisialisasi properti suara dari kelas Hewan dengan nilai "MBEEKURR", yang menggambarkan suara burung.
- Menginisialisasi properti berkembangBiak dari kelas Unggas dengan nilai "BERTELUR", yang menunjukkan bahwa burung berkembang biak dengan cara bertelur.
- Menginisialisasi properti jenis dengan nilai "MERPATI" dan ciri dengan nilai "BULU PUTIH", yang menggambarkan jenis burung dan ciri fisiknya.

Override cetak():

- Meng-override metode cetak() dari kelas Unggas untuk menampilkan informasi yang lebih spesifik mengenai burung.
- Informasi yang ditampilkan meliputi suara, cara berkembang biak, jenis, dan ciri khas burung.
- Setiap informasi ditampilkan menggunakan System.out.println().

Secara keseluruhan, kelas Burung memperluas kelas Unggas dengan menambahkan atribut jenis dan ciri, serta mengubah implementasi metode cetak() untuk menampilkan data yang lebih spesifik mengenai burung.

```

class Ayam extends Unggas {
    private String jenis;
    private String ciri;

    public Ayam(){
        suara = "POKK POKK PETOKK";
        berkembangBiak = "BERTELUR";
        this.jenis = "AYAM KAMPUNG";
        this.ciri = "BULU BERCORAK PUTIH HITAM";
    }

    public void cetak(){
        System.out.println("DATA HEWAN UNGGAS AYAM");
        System.out.println("SUARA : " + suara);
        System.out.println("BERKEMBANG BIAK : " + berkembangBiak);
        System.out.println("JENIS : " + jenis);
        System.out.println("CIRI : " + ciri);
    }
}

```

Kelas ini merupakan turunan (child class) dari Unggas yang menggambarkan objek Ayam.

Properti jenis dan ciri:

- jenis menyimpan jenis ayam, dengan tipe data String.
- ciri menyimpan ciri khas ayam, dengan tipe data String.

Konstruktor Ayam():

- Menginisialisasi properti suara dari kelas Hewan dengan nilai "POKK POKK PETOKK", yang menggambarkan suara ayam.
- Menginisialisasi properti berkembangBiak dari kelas Unggas dengan nilai "BERTELUR", yang menunjukkan bahwa ayam berkembang biak dengan cara bertelur.
- Menginisialisasi properti jenis dengan nilai "AYAM KAMPUNG" dan ciri dengan nilai "BULU BERCORAK PUTIH HITAM", yang menggambarkan jenis ayam dan ciri fisiknya.

Override cetak():

- Meng-override metode cetak() dari kelas Unggas untuk menampilkan informasi yang lebih spesifik mengenai ayam.
- Informasi yang ditampilkan meliputi suara, cara berkembang biak, jenis, dan ciri khas ayam.
- Setiap informasi ditampilkan menggunakan System.out.println().

Secara keseluruhan, kelas Ayam memperluas kelas Unggas dengan menambahkan atribut jenis dan ciri, serta mengubah implementasi metode cetak() untuk menampilkan data yang lebih spesifik mengenai ayam.

```
class Ikan extends Hewan {  
    protected String napas;  
  
    public Ikan(){  
        this.napas = "INSANG";  
    }  
  
    public void cetak(){  
        System.out.println("INSANG");  
    }  
}
```

Kelas ini merupakan turunan (child class) dari Hewan yang menggambarkan objek Ikan.

Properti napas:

- napas menyimpan cara ikan bernapas, dengan tipe data String.
- Properti ini diinisialisasi dengan nilai default "INSANG", yang menunjukkan bahwa ikan bernapas menggunakan insang.

Konstruktor Ikan():

- Menginisialisasi properti napas dengan nilai "INSANG", yang menggambarkan cara ikan bernapas.

Override cetak():

- Meng-override metode cetak() dari kelas Hewan untuk menampilkan informasi yang lebih spesifik mengenai ikan.
- Dalam metode ini, hanya mencetak informasi mengenai cara bernapas ikan, yaitu "INSANG", menggunakan System.out.println().

Secara keseluruhan, kelas Ikan memperluas kelas Hewan dengan menambahkan properti khusus yang menggambarkan cara ikan bernapas dan meng-override metode cetak() untuk menampilkan informasi ini.

```

class Gurami extends Ikan {
    private String Ciri;
    private String Berat;

    public Gurami(){
        suara = "(SUARA HATI)";
        napas = "INSANG";
        this.Ciri = "BERWARNA HITAM";
        this.Berat = "2";
    }

    public void cetak(){
        System.out.println("DATA HEWAN IKAN GURAMI");
        System.out.println("SUARA : " + suara);
        System.out.println("BERNAFAS : " + napas);
        System.out.println("CIRI : " + Ciri);
        System.out.println("BERAT : " + Berat + "KG");
    }
}

```

Kelas ini merupakan turunan (child class) dari Ikan yang lebih spesifik menggambarkan objek Gurami.

Properti Ciri dan Berat:

- Ciri menyimpan ciri fisik ikan gurami, dengan tipe data String.
- Berat menyimpan berat ikan gurami, dengan tipe data String.

Konstruktor Gurami():

- Menginisialisasi properti suara dari kelas Hewan dengan nilai "(SUARA HATI)", yang memberikan representasi suara gurami (meskipun ini lebih bersifat metaforis, karena ikan tidak bersuara seperti hewan lainnya).
- Menginisialisasi properti napas dari kelas Ikan dengan nilai "INSANG", yang menunjukkan bahwa gurami bernapas dengan insang.
- Menginisialisasi properti Ciri dengan nilai "BERWARNA HITAM", menggambarkan ciri fisik ikan gurami.
- Menginisialisasi properti Berat dengan nilai "2", yang menggambarkan berat ikan gurami dalam kilogram.

Override cetak():

- Meng-override metode cetak() dari kelas Ikan untuk menampilkan informasi yang lebih spesifik mengenai ikan gurami.
- Informasi yang ditampilkan meliputi suara, cara bernapas, ciri fisik, dan berat ikan gurami.
- Setiap informasi ditampilkan menggunakan System.out.println().

Secara keseluruhan, kelas Gurami memperluas kelas Ikan dengan menambahkan atribut Ciri dan Berat, serta mengubah implementasi metode cetak() untuk menampilkan data yang lebih spesifik mengenai ikan gurami.

```

class Lele extends Ikan {
    private String Ciri;
    private String Berat;

    public Lele(){
        suara = "---";
        napas = "INSANG";
        this.Ciri = "BERWARNA HITAM KEPUTIHAN";
        this.Berat = "3";
    }

    public void cetak(){
        System.out.println("DATA HEWAN IKAN LELE");
        System.out.println("SUARA : " + suara);
        System.out.println("BERNAFAS : " + napas);
        System.out.println("CIRI : " + Ciri);
        System.out.println("BERAT : " + Berat + "KG");
    }
}

```

Kelas ini merupakan turunan (child class) dari Ikan yang menggambarkan objek Lele.

Properti Ciri dan Berat:

- Ciri menyimpan ciri fisik ikan lele, dengan tipe data String.
- Berat menyimpan berat ikan lele, dengan tipe data String.

Konstruktor Lele():

- Menginisialisasi properti suara dari kelas Hewan dengan nilai "---", yang menggambarkan bahwa ikan lele tidak memiliki suara yang terdengar (diwakili dengan tanda ---).
- Menginisialisasi properti napas dari kelas Ikan dengan nilai "INSANG", yang menunjukkan bahwa lele bernapas dengan insang.
- Menginisialisasi properti Ciri dengan nilai "BERWARNA HITAM KEPUTIHAN", menggambarkan ciri fisik ikan lele.
- Menginisialisasi properti Berat dengan nilai "3", yang menggambarkan berat ikan lele dalam kilogram.

Override cetak():

- Meng-override metode cetak() dari kelas Ikan untuk menampilkan informasi yang lebih spesifik mengenai ikan lele.
- Informasi yang ditampilkan meliputi suara, cara bernapas, ciri fisik, dan berat ikan lele.
- Setiap informasi ditampilkan menggunakan System.out.println().

Secara keseluruhan, kelas Lele memperluas kelas Ikan dengan menambahkan atribut Ciri dan Berat, serta mengubah implementasi metode cetak() untuk menampilkan data yang lebih spesifik mengenai ikan lele.

=> Main Class

```
public class Main {  
    Run | Debug  
    public static void main(String[] args) {  
  
        System.out.print("\033[H\033[2J");  
  
        Sapi sp1 = new Sapi();  
        sp1.cetak();  
        System.out.println("");  
  
        Kambing kmb1 = new Kambing();  
        kmb1.cetak();  
        System.out.println("");  
  
        Burung brg1 = new Burung();  
        brg1.cetak();  
        System.out.println("");  
  
        Ayam aym1 = new Ayam();  
        aym1.cetak();  
        System.out.println("");  
  
        Gurami grm1 = new Gurami();  
        grm1.cetak();  
        System.out.println("");  
  
        Lele ll1 = new Lele();  
        ll1.cetak();  
        System.out.println("");  
    }  
}
```

Kelas Main ini digunakan untuk menjalankan program dan menguji semua kelas yang telah didefinisikan sebelumnya, seperti Sapi, Kambing, Burung, Ayam, Gurami, dan Lele.

Instansiasi objek dan pemanggilan metode cetak():

Untuk setiap kelas (seperti Sapi, Kambing, Burung, dll.), objek baru dibuat, dan metode cetak() dipanggil untuk menampilkan informasi terkait masing-masing objek.

- **Sapi:** Sapi sp1 = new Sapi();
Memanggil konstruktor dari kelas Sapi, yang kemudian menjalankan metode cetak() untuk menampilkan data tentang sapi.
- **Kambing:** Kambing kmb1 = new Kambing();
Sama halnya, objek Kambing dibuat dan cetak() dipanggil untuk menampilkan informasi kambing.
- **Burung:** Burung brg1 = new Burung();
Objek Burung dibuat dan informasi terkait burung ditampilkan.
- **Ayam:** Ayam aym1 = new Ayam();
Objek Ayam dibuat dan informasi ayam ditampilkan.
- **Gurami:** Gurami grm1 = new Gurami();
Objek Gurami dibuat dan informasi ikan gurami ditampilkan.
- **Lele:** Lele ll1 = new Lele();
Objek Lele dibuat dan informasi ikan lele ditampilkan.

=> Output

DATA HEWAN MAMALIA SAPI

SUARA : MOOOOHHH

BERKEMBANG BIAK : BERANAK

NAMA : AVRI

UMUR : 1TAHUN

DATA HEWAN MAMALIA KAMBING

SUARA : MBEEEEKK

BERKEMBANG BIAK : BERANAK

NAMA : TEGAR

UMUR : 2TAHUN

DATA HEWAN UNGGAS BURUNG

SUARA : MBEEKURR

BERKEMBANG BIAK : BERTELUR

JENIS : MERPATI

CIRI : BULU PUTIH

DATA HEWAN UNGGAS AYAM

SUARA : POKK POKK PETOKK

BERKEMBANG BIAK : BERTELUR

JENIS : AYAM KAMPUNG

CIRI : BULU BERCORAK PUTIH HITAM

DATA HEWAN IKAN GURAMI

SUARA : (SUARA HATI)

BERNAFAS : INSANG

CIRI : BERWARNA HITAM

BERAT : 2KG

DATA HEWAN IKAN LELE

SUARA : ---

BERNAFAS : INSANG

CIRI : BERWARNA HITAM KEPUTIHAN

BERAT : 3KG