**Program inheritance**

**Contoh 1**

// Kelas utama

class Hewan {

// Properti

String jenis;

int umur;

// Konstruktor

public Hewan(String jenis, int umur) {

this.jenis = jenis;

this.umur = umur;

}

// Metode

public void bersuara() {

System.out.println("Hewan ini bersuara.");

}

}

// Subkelas yang mewarisi dari kelas Hewan

class Kucing extends Hewan {

// Konstruktor

public Kucing(String jenis, int umur) {

super(jenis, umur); // Memanggil konstruktor kelas induk (Hewan)

}

// Metode khusus untuk kelas Kucing

public void bersuara() {

System.out.println("Meong!");

}

// Metode tambahan untuk kelas Kucing

public void tidur() {

System.out.println("Kucing sedang tidur.");

}

}

// Subkelas lain yang mewarisi dari kelas Hewan

class Anjing extends Hewan {

// Konstruktor

public Anjing(String jenis, int umur) {

super(jenis, umur); // Memanggil konstruktor kelas induk (Hewan)

}

// Metode khusus untuk kelas Anjing

public void bersuara() {

System.out.println("Guk guk!");

}

// Metode tambahan untuk kelas Anjing

public void berlari() {

System.out.println("Anjing sedang berlari.");

}

}

// Kelas utama untuk menjalankan program

public class Main {

public static void main(String[] args) {

// Membuat objek kucing

Kucing kucing = new Kucing("Persia", 3);

System.out.println("Kucing jenis " + kucing.jenis + " berumur " + kucing.umur + " tahun.");

kucing.bersuara(); // Memanggil metode dari subkelas Kucing

kucing.tidur(); // Memanggil metode dari subkelas Kucing

// Membuat objek anjing

Anjing anjing = new Anjing("Bulldog", 5);

System.out.println("Anjing jenis " + anjing.jenis + " berumur " + anjing.umur + " tahun.");

anjing.bersuara(); // Memanggil metode dari subkelas Anjing

anjing.berlari(); // Memanggil metode dari subkelas Anjing

}

}

Output

Kucing jenis Persia berumur 3 tahun.

Meong!

Kucing sedang tidur.

Anjing jenis Bulldog berumur 5 tahun.

Guk guk!

Anjing sedang berlari.

Contoh 2 .

// Kelas utama

class Shape {

// Metode

public void draw() {

System.out.println("Menggambar bentuk.");

}

}

// Subkelas yang mewarisi dari kelas Shape

class Circle extends Shape {

// Metode khusus untuk kelas Circle

public void draw() {

System.out.println("Menggambar lingkaran.");

}

}

// Subkelas lain yang mewarisi dari kelas Shape

class Rectangle extends Shape {

// Metode khusus untuk kelas Rectangle

public void draw() {

System.out.println("Menggambar persegi panjang.");

}

}

// Kelas utama untuk menjalankan program

public class Main {

public static void main(String[] args) {

// Membuat objek lingkaran

Circle lingkaran = new Circle();

System.out.println("Membuat objek lingkaran:");

lingkaran.draw(); // Memanggil metode dari subkelas Circle

// Membuat objek persegi panjang

Rectangle persegiPanjang = new Rectangle();

System.out.println("\nMembuat objek persegi panjang:");

persegiPanjang.draw(); // Memanggil metode dari subkelas Rectangle

}

}

Output :

Membuat objek lingkaran:

Menggambar lingkaran.

Membuat objek persegi panjang:

Menggambar persegi panjang.

Contoh 3.

// Kelas utama

class Shape {

// Metode

public void display() {

System.out.println("Ini adalah bentuk.");

}

// Metode abstrak untuk menghitung luas

public double calculateArea() {

return 0.0;

}

// Metode abstrak untuk menghitung keliling

public double calculatePerimeter() {

return 0.0;

}

}

// Subkelas yang mewarisi dari kelas Shape

class Rectangle extends Shape {

// Properti

double width;

double height;

// Konstruktor

public Rectangle(double width, double height) {

this.width = width;

this.height = height;

}

// Override metode display() dari kelas induk

@Override

public void display() {

System.out.println("Ini adalah persegi panjang.");

}

// Override metode calculateArea() dari kelas induk untuk menghitung luas persegi panjang

@Override

public double calculateArea() {

return width \* height;

}

// Override metode calculatePerimeter() dari kelas induk untuk menghitung keliling persegi panjang

@Override

public double calculatePerimeter() {

return 2 \* (width + height);

}

}

// Subkelas lain yang mewarisi dari kelas Shape

class Circle extends Shape {

// Properti

double radius;

final double PI = 3.14; // Nilai Pi

// Konstruktor

public Circle(double radius) {

this.radius = radius;

}

// Override metode display() dari kelas induk

@Override

public void display() {

System.out.println("Ini adalah lingkaran.");

}

// Override metode calculateArea() dari kelas induk untuk menghitung luas lingkaran

@Override

public double calculateArea() {

return PI \* radius \* radius;

}

// Override metode calculatePerimeter() dari kelas induk untuk menghitung keliling lingkaran

@Override

public double calculatePerimeter() {

return 2 \* PI \* radius;

}

}

// Kelas utama untuk menjalankan program

public class Main {

public static void main(String[] args) {

// Membuat objek persegi panjang

Rectangle rectangle = new Rectangle(5.0, 3.0);

rectangle.display(); // Memanggil metode display() dari subkelas Rectangle

System.out.println("Luas: " + rectangle.calculateArea());

System.out.println("Keliling: " + rectangle.calculatePerimeter());

// Membuat objek lingkaran

Circle circle = new Circle(4.0);

circle.display(); // Memanggil metode display() dari subkelas Circle

System.out.println("Luas: " + circle.calculateArea());

System.out.println("Keliling: " + circle.calculatePerimeter());

}

}

Output :

Ini adalah persegi panjang.

Luas: 15.0

Keliling: 16.0

Ini adalah lingkaran.

Luas: 50.24

Keliling: 25.12