**TUGAS INHARINTANCE PEMROGRAMAN BERBASIS OBJECT**

Disusun untuk Memenuhi Tugas Mata Kuliah Pemrograman Berbasis Object yang Dibimbing oleh YOGA DWITYA PRAMUDITA, S.Kom., M.Cs.

****

Disusun oleh:

Achmad Farid Alfa Waid (190411100073)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**2020**

1. **CODE PROGRAM**

**Hewan.java**

public class Hewan {

public String swimming (String renang) {

return renang;

}

public String flying (String terbang) {

return terbang;

}

public String walking (String jalan) {

return jalan;

}

public String running (String lari) {

return lari;

}

}

class Elang extends Hewan {

public void tingkah\_laku(){

System.out.println("--------- HEWAN ELANG ---------");

System.out.println("Elang bernafas : " + this.bernafas("wushhh"));

System.out.println("Suara Elang terbang : " + super.flying("Wing - wing"));

System.out.println("Suara Elang : " + this.bersuara("Kwuaikkk"));

System.out.println("Elang berkembangbiak dengan cara : " + this.berkembangbiak("Bertelur"));

System.out.println("Elang makan : " + this.makan("Daging"));

}

public String bernafas(String nafas){

return nafas;

}

public String bersuara(String suara){

return suara;

}

public String berkembangbiak(String KB){

return KB;

}

public String makan(String makanan){

return makanan;

}

}

class Bebek extends Hewan{

public void tingkah\_laku(){

System.out.println("--------- HEWAN BEBEK ---------");

System.out.println("Bebek bernafas : " + this.bernafas("huimmm"));

System.out.println("Suara Bebek berenang : " + super.swimming("Kecipak - kecipuk"));

System.out.println("Suara Bebek : " + this.bersuara("Wekk wekkk"));

System.out.println("Bebek berkembangbiak dengan cara : " + this.berkembangbiak("Bertelur"));

System.out.println("Bebek makan : " + this.makan("Dedak"));

}

public String bernafas(String nafas){

return nafas;

}

public String bersuara(String suara){

return suara;

}

public String berkembangbiak(String KB){

return KB;

}

public String makan(String makanan){

return makanan;

}

}

class Sapi extends Hewan {

public void tingkah\_laku(){

System.out.println("--------- HEWAN SAPI ---------");

System.out.println("Sapi bernafas : " + this.bernafas("ehmmm"));

System.out.println("Suara Sapi berjalan : " + super.walking("Klutak - Klutuk"));

System.out.println("Suara Sapi : " + this.bersuara("Mooooo"));

System.out.println("Sapi berkembangbiak dengan cara : " + this.berkembangbiak("Beranak / Melahirkan"));

System.out.println("Sapi makan : " + this.makan("Rumput"));

}

public String bernafas(String nafas){

return nafas;

}

public String bersuara(String suara){

return suara;

}

public String berkembangbiak(String KB){

return KB;

}

public String makan(String makanan){

return makanan;

}

}

class Domba extends Hewan {

public void tingkah\_laku(){

System.out.println("--------- HEWAN DOMBA ---------");

System.out.println("Domba bernafas : " + this.bernafas("werreee"));

System.out.println("Suara Domba berlari : " + super.running("gruduk - gruduk"));

System.out.println("Suara Domba : " + this.bersuara("Mbekkkk"));

System.out.println("Domba berkembangbiak dengan cara : " + this.berkembangbiak("Beranak / Melahirkan"));

System.out.println("Domba makan : " + this.makan("Rumput"));

}

public String bernafas(String nafas){

return nafas;

}

public String bersuara(String suara){

return suara;

}

public String berkembangbiak(String KB){

return KB;

}

public String makan(String makanan){

return makanan;

}

}

Membuat sebuah public class yang bernama Hewan, class tersebut merupakan superclass. Di dalam class tersebut tedapat beberapa method, yaitu method swimming, flying, walking, dan running.

Terdapat 5 subclass yaitu class Elang, Bebek, Sapi, Domba. Masing – masing class tersebut mewarisi class Hewan, dan masing – masing dari class tersebut memiliki method bernafas, bersuara, berkembangbiak, dan makan. Kemudian membuat method tingkah\_laku untuk menampilkan beberapa method yang sudah disebutkan di atas, untuk mengakses nilai method dari subclass sendiri menggunakan syntax this, sedangkan untuk superclass mengunakan syntax super

**Uji.java**

public class Uji extends Hewan {

public static void main(String [] args){

Elang elang = new Elang();

elang.tingkah\_laku();

Bebek bebek = new Bebek();

bebek.tingkah\_laku();

Sapi sapi = new Sapi();

sapi.tingkah\_laku();

Domba domba = new Domba();

domba.tingkah\_laku();

}

}

Membuat sebuah public class yang bernama Uji, class ini merupakan warisan dari class Hewan. Kemudian membuat object baru dari subclass yang ada di class hewan, dan mengakses method tingkah\_laku yang ada di masing – masing object untuk ditampilakan di dalam terminal.

**Bergerak.java**

public interface Bergerak {

public void maju(int satuan);

public void mundur(int satuan);

public void kanan(int satuan);

public void kiri(int satuan);

}

class Kerbau implements Bergerak {

public void maju(int satuan){

System.out.println("--------- HEWAN KERBAU ---------");

System.out.println("Kerbau bergerak maju " + satuan + " langkah");

}

public void mundur(int satuan){

System.out.println("Kerbau bergerak mundur " + satuan + " langkah");

}

public void kanan(int satuan){

System.out.println("Kerbau bergerak ke kanan " + satuan + " langkah");

}

public void kiri(int satuan){

System.out.println("Kerbau bergerak ke kiri " + satuan + " langkah");

}

public void makan(String makanan){

System.out.println("Kerbau memakan : " + makanan);

}

}

class Mobil implements Bergerak {

public void maju(int satuan){

System.out.println("--------- MOBIL ---------");

System.out.println("Mobil bergerak maju " + satuan + " km/jam");

}

public void mundur(int satuan){

System.out.println("Mobil bergerak mundur " + satuan + " km/jam");

}

public void kanan(int satuan){

System.out.println("Mobil bergerak ke kanan " + satuan + " km/jam");

}

public void kiri(int satuan){

System.out.println("Mobil bergerak ke kiri " + satuan + " km/jam");

}

public void isiBensin (int satuan){

System.out.println("Mobil mengisi bensin sebanyak " + satuan + " liter");

}

}

class Robot implements Bergerak {

public void maju(int satuan){

System.out.println("--------- ROBOT ---------");

System.out.println("Robot bergerak maju " + satuan + " m/detik");

}

public void mundur(int satuan){

System.out.println("Robot bergerak mundur " + satuan + " m/detik");

}

public void kanan(int satuan){

System.out.println("Robot bergerak ke kanan " + satuan + " m/detik");

}

public void kiri(int satuan){

System.out.println("Robot bergerak ke kiri " + satuan + " m/detik");

}

public void baterai(String daya){

System.out.println("Robot memakai baterai bersumber daya " + daya);

}

}

Membuat sebuah public class interface yang bernama Bergerak, class tersebut merupakan superclass. Di dalam class tersebut tedapat beberapa method, yaitu method maju, mundur, kanan, dan kiri.

Selanjutnya terdapat tiga subclass yaitu class Kerbau, Mobil, dan Robot. Masing – masing class tersebut mewarisi superclass Bergerak. Sehingga dapat mengakses method maju, mundur, kanan, dan kiri. Kemudian pada subclass Kerbau terdapat method makan, untuk subclass Mobil terdapat method isiBensin, dan pada subclass Robot terdapat method baterai

**Uji2.java**

public class Uji2 {

public static void main(String[] args) {

Kerbau kerbau = new Kerbau();

kerbau.maju(5);

kerbau.mundur(10);

kerbau.kanan(15);

kerbau.kiri(20);

kerbau.makan("Rumput");

Mobil mobil = new Mobil();

mobil.maju(3);

mobil.mundur(6);

mobil.kanan(9);

mobil.kiri(12);

mobil.isiBensin(5);

Robot robot = new Robot();

robot.maju(2);

robot.mundur(4);

robot.kanan(6);

robot.kiri(8);

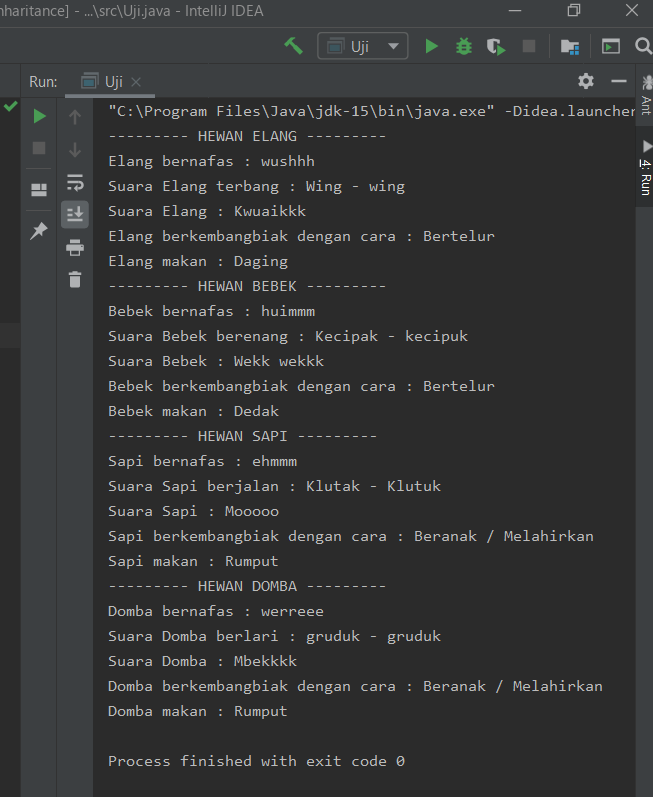
robot.baterai("listrik");

}

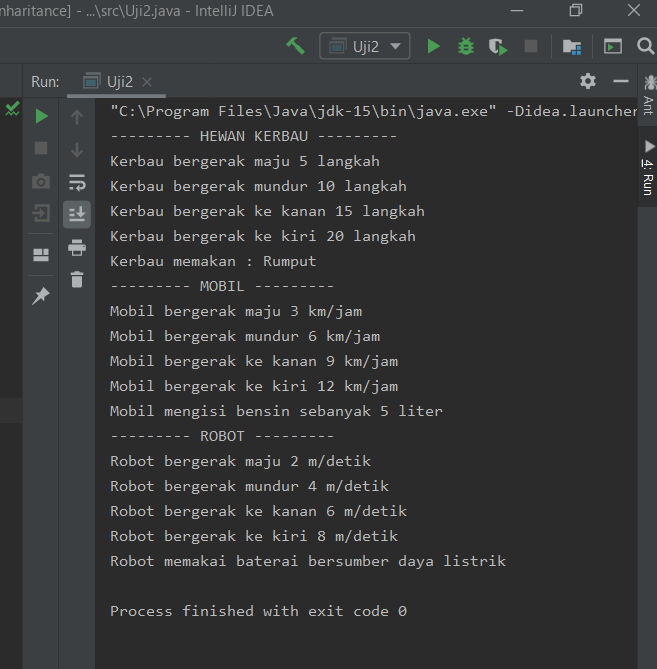
}

Membuat sebuah public class yang bernama Uji, pada class ini akan mebuat object baru dari masing – masing subclass yang ada di class Bergerak, kemudian mengakses method – method yang ada di masing – masing class tersebut untuk ditampilkan di dalam terminal.

1. **HASIL RUNNING PROGRAM**
2. **Uji.java**

****

1. **Uji2.java**

****