

**LAPORAN PRAKTIKUM
PEMROGRAMAN II
MODUL 4**



PEWARISAN/INHERITANCE & DIAGRAM KELAS

Oleh:

Ryan Muhammad Irfan NIM. 2210817310013

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
NOVEMBER 2023**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN II
MODUL 4

Laporan Praktikum Pemrograman II Modul 4: Pewarisan/Inheritance & Diagram Kelas ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman II. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Ryan Muhammad Irfan
NIM : 2210817310013

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Bachrul Uluum
NIM. 2010817210025

Andreyan Rizky Baskara, S.Kom., M.Kom.
NIP. 19930703 201903 1 011

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code	7
B. Output Program.....	8
C. Pembahasan.....	8
D. Tautan GIT.....	12
SOAL 2.....	13
A. Source Code	15
B. Output Program.....	17
C. Pembahasan.....	18
D. Tautan GIT.....	29

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1 (1).....	8
Gambar 2. Screenshot Hasil Jawaban Soal 2 (1).....	17
Gambar 3. Screenshot Hasil Jawaban Soal 2 (2).....	18

DAFTAR TABEL

Tabel 1. Soal 1 (Output)	6
Tabel 2. Source Code Soal 1(HewanPeliharaan.java)	7
Tabel 3. Source Code Soal 1 (Main.java).....	7
Tabel 4. Soal 2 (Output)	14
Tabel 5. Source Code Soal 2 (HewanPeliharaan.java)	15
Tabel 6. Source Code Soal 2 (Kucing.java)	15
Tabel 7. Source Code Soal 2 (Anjing.java)	16
Tabel 8. Source Code Soal 2 (Main.java).....	17

SOAL 1

Diberikan class diagram seperti berikut: (isi program harus sesuai dengan class diagram)

HewanPeliharaan
- nama : String - ras : String
+ HewanPeliharaan(r: String, n: String) + display(): void

Implementasikan class diagram diatas menjadi program menggunakan Bahasa pemrograman java dengan ketentuan seperti berikut:

- Inisiasi objek **harus** menggunakan konstruktor.
- Output ditampilkan **dengan hanya** memanggil method display.
- Tidak boleh ada fungsi mencetak di class main.

Input
Nama Hewan Peliharaan: Leo Ras: Kucing Anggora
Output
Detail Hewan Peliharaan: Nama hewan peliharaanku adalah : Leo Dengan ras : Kucing Anggora

Tabel 1. Soal 1 (Output)

Simpan coding anda dengan nama package: **soal1**

A. Source Code

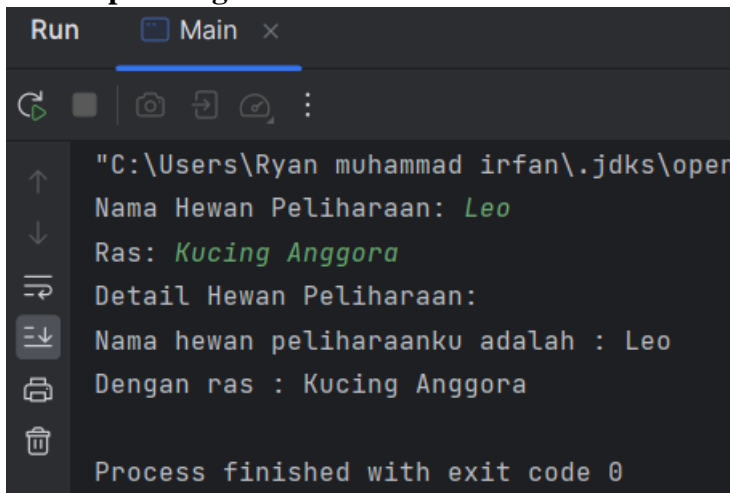
HewanPeliharaan.java	
1	package soall1;
2	
3	import java.util.Scanner;
4	
5	public class HewanPeliharaan {
6	private String nama;
7	private String ras;
8	Scanner userinput = new Scanner(System.in);
9	
10	public HewanPeliharaan(String r, String n){
11	this.ras = r;
12	this.nama = n;
13	}
14	
15	public void display(){
16	System.out.print("Nama Hewan Peliharaan: ");
17	nama = userinput.nextLine();
18	System.out.print("Ras: ");
19	ras = userinput.nextLine();
20	System.out.println("Detail Hewan Peliharaan:");
21	System.out.println("Nama hewan peliharaanku
22	adalah : " + nama);
23	System.out.println("Dengan ras : " + ras);
24	}
25	}

Tabel 2. Source Code Soal 1(HewanPeliharaan.java)

Main.java	
1	package soall1;
2	
3	public class Main {
4	public static void main(String[] args) {
5	HewanPeliharaan hewan = new HewanPeliharaan(null,null);
6	hewan.display();
7	}
8	}

Tabel 3. Source Code Soal 1 (Main.java)

B. Output Program



```
Run Main x
"C:\Users\Ryan muhammad irfan\.jdk\open
Nama Hewan Peliharaan: Leo
Ras: Kucing Anggora
Detail Hewan Peliharaan:
Nama hewan peliharaanku adalah : Leo
Dengan ras : Kucing Anggora
Process finished with exit code 0
```

Gambar 1. Screenshot Hasil Jawaban Soal 1 (1)

C. Pembahasan

HewanPeliharaan.java

Pada baris [1], `package soal1;`

“package soal1” berfungsi untuk mendeklarasikan package dengan nama soal1. Class HewanPeliharaan berada dalam package soal1.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3], `import java.util.Scanner;`

Mengimpor class scanner yang terdapat di dalam package java.util. Berfungsi untuk membaca input dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [5-24], `public class HewanPeliharaan`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama HewanPeliharaan.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [6], `private String nama;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“String” merupakan tipe data dari atribut nama.

“nama” merupakan atribut dari sebuah class HewanPeliharaan.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [7], `private String ras;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“String” merupakan tipe data dari atribut ras.

“ras” merupakan atribut dari sebuah class HewanPeliharaan.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [8], `Scanner userInput = new Scanner(System.in);`

Berfungsi untuk membuat sebuah objek dari class Scanner atau dalam hal ini variabel userInput akan berisi object dari Scanner class dengan cara memanggil konstruktor Scanner(System.in). System in adalah input stream yang digunakan untuk menerima input dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [10-13], `public HewanPeliharaan(String r, String n){...}`

“public” berfungsi sebagai penanda bahwa constructor dapat diakses dari class lain.

“HewanPeliharaan(String r, String n){...}” merupakan sebuah constructor dari class HewanPeliharaan. Constructor ini menerima parameter String r, String n.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [10], `this.ras = r;`

“this.ras = r” berfungsi untuk menginisialisasi atribut “ras” dengan nilai yang diterima oleh parameter “r”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [11], `this.nama = n;`

“this.nama = n” berfungsi untuk menginisialisasi atribut “nama” dengan nilai yang diterima oleh parameter “n”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [15-23], `public void display(){...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari class lain.

“void” berfungsi menyatakan method tersebut tidak dapat mengembalikan nilai.

“display()” merupakan sebuah method dengan nama display. Dapat dipanggil pada pada class lain.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [16], `System.out.print("Nama Hewan Peliharaan: ");`

“System.out.print(...)” berfungsi untuk mencetak angka atau karakter. Dalam baris ini mencetak " Nama Hewan Peliharaan: ”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [17], `nama = userInput.nextLine();`

“nama” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextLine()`. <nama variabel> = <nilai variabel>.

“userInput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“nextLine()” berfungsi untuk membaca nilai String dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [18], `System.out.print("Ras: ");`

“System.out.print(...)” berfungsi untuk mencetak angka atau karakter. Dalam baris ini mencetak "Ras: ”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [19], `Ras = userInput.nextLine();`

“Ras” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextLine()`. <nama variabel> = <nilai variabel>.

“userInput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“nextLine()” berfungsi untuk membaca nilai String dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [20], `System.out.println("Detail Hewan Peliharaan: ");`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Detail Hewan Peliharaan: ”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [21], `System.out.println("Nama hewan peliharaanku adalah : ");`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Nama hewan peliharaanku adalah : ” + value nama.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [22], `System.out.println("Dengan ras : ");`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Dengan ras : ” + value ras.

“;” berfungsi untuk mengakhiri sebuah perintah.

Main.java

Pada baris [1], `package soal1;`

“package soal1” berfungsi untuk mendeklarasikan package dengan nama soal1. Class Main berada dalam package soal1.

Pada baris [3-8], `public class Main`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama Main.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [4-7], `public static void main(String[] args)`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“static” berfungsi membuat suatu method tanpa perlu melakukan instansiasi terlebih dahulu.

“void” berfungsi untuk tidak mengembalikan nilai apapun.

“main” merupakan nama fungsi yang digunakan oleh java sebagai awal masuk ke program.

“String[] args” berfungsi sebagai parameter yang diperlukan oleh fungsi main. Parameter ini adalah array dari argument perintah yang bisa diteruskan ke program java.

Pada baris [5], `HewanPeliharaan hewan = new HewanPeliharaan(null, null);`

“HewanPeliharaan” merupakan sebuah class.

“hewan” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class HewanPeliharaan.

“new HewanPeliharaan” membuat objek baru dari kelas HewanPeliharaan.

“(null, null)” berfungsi untuk mengisi value atribut dari parameter yang sudah ditentukan. Pada baris ini mengisi value atribut dengan parameter n: null, r: null.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [6], `hewan.display();`

“hewan.display” berfungsi untuk memanggil method display pada objek “hewan” dari class HewanPeliharaan.

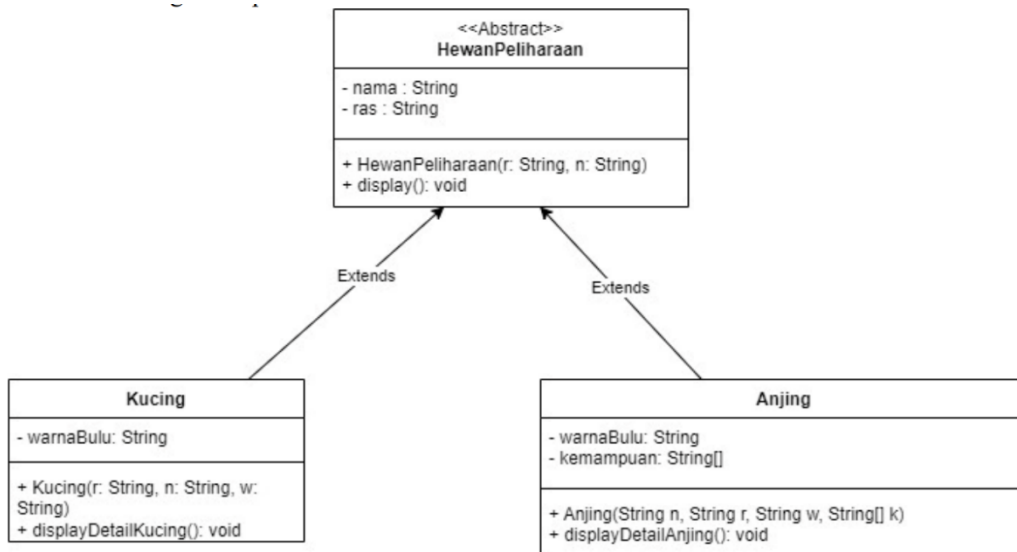
“;” berfungsi untuk mengakhiri sebuah perintah.

D. Tautan GIT

<https://github.com/ryanmi04/Praktikum-Pemrograman-2-Paralel-1/tree/main/PRAKTIKUM%204/src/soal1>

SOAL 2

Diberikan class diagram seperti berikut:



Implementasikan class diagram diatas menjadi program menggunakan Bahasa pemrograman java dengan ketentuan seperti berikut:

- Inisiasi objek harus menggunakan konstruktor.
- Isi baris program harus sesuai dengan apa yang digambarkan di dalam kelas diagram.
- Menampilkan detail dari class Kucing maupun Anjing harus melalui method yang ada, tidak boleh menggunakan fungsi cetak dari java
- Isi detail dari class Kucing dan Anjing harus memanggil method display pada parent class terlebih dahulu.

Input
Pilih jenis hewan yang ingin diinputkan: 1 = Kucing 2 = Anjing Masukkan pilihan: 1 Nama hewan peliharaan: lolo Ras: Anggora
Output
Detail Hewan Peliharaan: Nama hewan peliharaanku adalah : lolo

Dengan ras : Anggora Memiliki warna bulu : abu-abu
Input
Pilih jenis hewan yang ingin diinputkan: 1 = Kucing 2 = Anjing Masukkan pilihan: 2 Nama hewan peliharaan: andi Ras: cihuahua Warna Bulu: putih Kemampuan : berenang, berjabat tangan, melompat
Output
Detail Hewan Peliharaan: Nama hewan peliharaanku adalah : andi Dengan ras : cihuahua Memiliki warna bulu : putih Memiliki kemampuan : berenang berjabat tangan melompat

Tabel 4. Soal 2 (Output)

Simpan coding anda dengan nama package: **soal2**

A. Source Code

HewanPeliharaan.java	
1	package soal2;
2	
3	public class HewanPeliharaan {
4	private String nama;
5	private String ras;
6	
7	public HewanPeliharaan(String r, String n){
8	this.ras = r;
9	this.nama = n;
10	}
11	
12	public void display(){
13	System.out.println("Detail Hewan Peliharaan:");
14	System.out.println("Nama hewan peliharaanku
15	adalah: " + nama);
16	System.out.println("Dengan Ras: " + ras);
17	}

Tabel 5. Source Code Soal 2 (HewanPeliharaan.java)

Kucing.java	
1	package soal2;
2	
3	public class Kucing extends HewanPeliharaan {
4	private String warnaBulu;
5	public Kucing(String r, String n, String w) {
6	super(r, n);
7	this.warnaBulu = w;
8	}
9	
10	public void displayDetailKucing(){
11	super.display();
12	System.out.println("Memiliki warna bulu: " +
13	warnaBulu);
14	}

Tabel 6. Source Code Soal 2 (Kucing.java)

Anjing.java	
1	package soal2;
2	
3	public class Anjing extends HewanPeliharaan {
4	private String warnaBulu;
5	private String[] kemampuan;
6	
7	public Anjing(String n, String r, String w, String[] k) {
8	super(n, r);

9	this.warnaBulu = w;
10	this.kemampuan = k;
11	}
12	
13	public void displayDetailAnjing(){
14	super.display();
15	System.out.println("Memiliki warna bulu : " + warnaBulu);
16	System.out.print("Memiliki kemampuan :");
17	for(String kemampuanAnjing : kemampuan){
18	System.out.print(" " + kemampuanAnjing);
19	}
20	
21	}
22	}

Tabel 7. Source Code Soal 2 (Anjing.java)

Main.java	
1	package soal2;
2	
3	import java.util.Scanner;
4	
5	public class Main {
6	public static void main(String[] args) {
7	Scanner userInput = new Scanner(System.in);
8	String nama, ras, warnabulu, kemampuanInput;
9	
10	System.out.println("Pilih jenis hewan yang ingin diinputkan:");
11	System.out.println("1 = Kucing");
12	System.out.println("2 = Anjing");
13	System.out.print("Masukan pilihan: ");
14	int pilihan = userInput.nextInt();
15	userInput.nextLine();
16	
17	switch (pilihan){
18	case 1:
19	System.out.print("Nama hewan peliharaan: ");
20	nama = userInput.nextLine();
21	System.out.print("Ras: ");
22	ras = userInput.nextLine();
23	System.out.print("Warna Bulu: ");
24	warnabulu = userInput.nextLine();
25	Kucing k = new Kucing(ras, nama, warnabulu);
26	k.displayDetailKucing();
27	break;
28	
29	case 2:
30	System.out.print("Nama hewan peliharaan: ");
31	nama = userInput.nextLine();
32	System.out.print("Ras: ");
33	ras = userInput.nextLine();
34	System.out.print("Warna Bulu: ");
35	warnabulu = userInput.nextLine();
36	System.out.print("Kemampuan : ");
37	kemampuanInput = userInput.nextLine();

38	String[] kemampuan = kemampuanInput.split(", ");
39	Anjing a = new Anjing(nama, ras, warnabulu,
	kemampuan);
40	a.displayDetailAnjing();
41	break;
42	
43	default:
44	System.out.println("Pilihan tidak valid");
45	}
46	}
47	}

Tabel 8. Source Code Soal 2 (Main.java)

B. Output Program

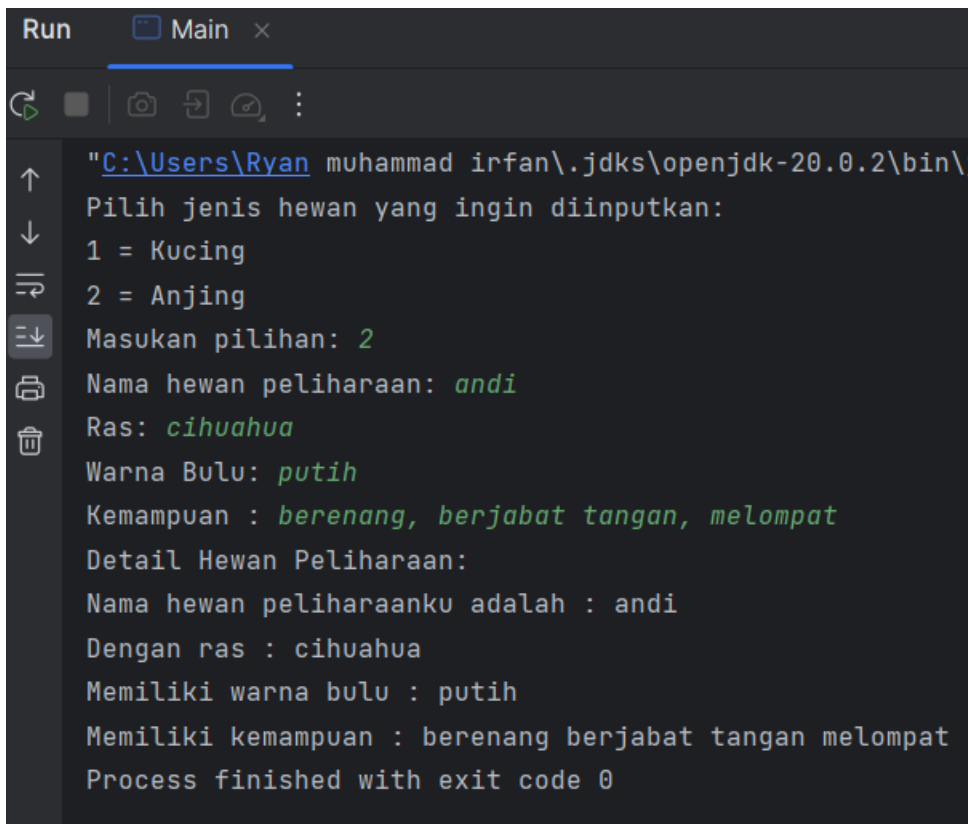
```

Run Main x
"C:\Users\Ryan muhammad irfan\.jdk\openjdk-20
Pilih jenis hewan yang ingin diinputkan:
1 = Kucing
2 = Anjing
Masukan pilihan: 1
Nama hewan peliharaan: lolo
Ras: Anggora
Warna Bulu: abu-abu
Detail Hewan Peliharaan:
Nama hewan peliharaanku adalah : lolo
Dengan ras : Anggora
Memiliki warna bulu : abu-abu

Process finished with exit code 0

```

Gambar 2. Screenshot Hasil Jawaban Soal 2 (1)



```
Run Main x
"C:\Users\Ryan muhammad irfan\.jdk\openjdk-20.0.2\bin\j
Pilih jenis hewan yang ingin diinputkan:
1 = Kucing
2 = Anjing
Masukan pilihan: 2
Nama hewan peliharaan: andi
Ras: cihuahua
Warna Bulu: putih
Kemampuan : berenang, berjabat tangan, melompat
Detail Hewan Peliharaan:
Nama hewan peliharaanku adalah : andi
Dengan ras : cihuahua
Memiliki warna bulu : putih
Memiliki kemampuan : berenang berjabat tangan melompat
Process finished with exit code 0
```

Gambar 3. Screenshot Hasil Jawaban Soal 2 (2)

C. Pembahasan

HewanPeliharaan.java

Pada baris [1], `package soal2;`

“package soal2” berfungsi untuk mendeklarasikan package dengan nama soal2. Class HewanPeliharaan berada dalam package soal2.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3-17], `public class HewanPeliharaan`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama HewanPeliharaan.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [4], `private String nama;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“String” merupakan tipe data dari atribut nama.

“nama” merupakan atribut dari sebuah class HewanPeliharaan.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [5], `private String ras;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“String” merupakan tipe data dari atribut ras.

“ras” merupakan atribut dari sebuah class HewanPeliharaan.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [7-10], `public HewanPeliharaan(String r, String n){...}`

“public” berfungsi sebagai penanda bahwa constructor dapat diakses dari class lain.

“HewanPeliharaan(String r, String n){...}” merupakan sebuah constructor dari class HewanPeliharaan. Constructor ini menerima parameter String r, String n.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [8], `this.ras = r;`

“this.ras = r” berfungsi untuk menginisialisasi atribut “ras” dengan nilai yang diterima oleh parameter “r”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [9], `this.nama = n;`

“this.nama = n” berfungsi untuk menginisialisasi atribut “nama” dengan nilai yang diterima oleh parameter “n”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [12-16], `public void display(){...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari class lain.

“void” berfungsi menyatakan method tersebut tidak dapat mengembalikan nilai.

“display()” merupakan sebuah method dengan nama display. Dapat dipanggil pada pada class lain.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [13], `System.out.println("Detail Hewan Peliharaan: ");`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Detail Hewan Peliharaan : ”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [14], `System.out.println("Nama hewan peliharaanku adalah : ");`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Nama hewan peliharaanku adalah : " + value nama.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [15], `System.out.println("Dengan ras : ");`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Dengan ras : " + value ras.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Kucing.java

Pada baris [1], `package soal2;`

“`package soal2`” berfungsi untuk mendeklarasikan package dengan nama soal2. Class Kucing berada dalam package soal2.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3-14], `public class Kucing extends HewanPeliharaan {`

“`public`” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“`class`” berfungsi untuk membuat class yang dalam baris ini diberi nama Kucing.

“`extends HewanPeliharaan`” berfungsi untuk mewariskan method atau atribut dari kelas induk. Dalam hal ini kelas induknya HewanPeliharaan.

“`{...}`” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [4], `private String warnaBulu;`

“`private`” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“`String`” merupakan tipe data dari atribut warnaBulu.

“`warnaBulu`” merupakan atribut dari sebuah class Kucing.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [5-8], `public Kucing(String r, String n, String w){...}`

“`public`” berfungsi sebagai penanda bahwa constructor dapat diakses dari class lain.

“Kucing(String r, String n, String w){...}” merupakan sebuah constructor dari class Kucing. Constructor ini menerima parameter String r, String n, String w.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [6], `super(r, n);`

“super(r, n)” berfungsi untuk mengakses konstruktor dari kelas induk

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [7], `this.warnaBulu = w;`

“this.warnaBulu = w” berfungsi untuk menginisialisasi atribut “warnaBulu” dengan nilai yang diterima oleh parameter “w”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [10-13], `public void displayDetailKucing(){...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari class lain.

“void” berfungsi menyatakan method tersebut tidak dapat mengembalikan nilai.

“displayDetailKucing()” merupakan sebuah method dengan nama displayDetailKucing. Dapat dipanggil pada pada class lain.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [11], `super.display();`

“super.display()” berfungsi untuk memanggil method display dengan mengakses kelas induk.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [12], `System.out.println("Memiliki warna bulu : " + warnaBulu);`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Memiliki warna bulu : " + value warnaBulu.

“;” berfungsi untuk mengakhiri sebuah perintah.

Anjing.java

Pada baris [1], `package soal2;`

“package soal2” berfungsi untuk mendeklarasikan package dengan nama soal2. Class Anjing berada dalam package soal2.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3-22], `public class public class Anjing extends HewanPeliharaan {`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama Anjing.

“extends HewanPeliharaan” berfungsi untuk mewariskan method atau atribut dari kelas induk. Dalam hal ini kelas induknya HewanPeliharaan.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [4], `private String warnaBulu;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“String” merupakan tipe data dari atribut warnaBulu.

“warnaBulu” merupakan atribut dari sebuah class Kucing.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [5], `private String[] kemampuan;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“String[]” merupakan tipe data dari atribut kemampuan. Atribut ini sebuah array.

“kemampuan” merupakan atribut dari sebuah class Anjing.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [7-11], `public Anjing(String n, String r, String w, String[] k){...}`

“public” berfungsi sebagai penanda bahwa constructor dapat diakses dari class lain.

“Anjing(String n, String r, String w, String[] k){...}” merupakan sebuah constructor dari class Anjing. Constructor ini menerima parameter String n, String r, String w, String[] k.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [8], `super(n, r);`

“super(n, r)” berfungsi untuk mengakses konstruktor dari kelas induk

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [9], `this.warnaBulu = w;`

“this.warnaBulu = w” berfungsi untuk menginisialisasi atribut “warnaBulu” dengan nilai yang diterima oleh parameter “w”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [10], `this.kemampuan = k;`

“`this.kemampuan = k`” berfungsi untuk menginisialisasi atribut “kemampuan” dengan nilai yang diterima oleh parameter “k”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [13-21], `public void displayDetailAnjing() {...}`

“`public`” berfungsi sebagai penanda bahwa method ini dapat diakses dari class lain.

“`void`” berfungsi menyatakan method tersebut tidak dapat mengembalikan nilai.

“`displayDetailAnjing()`” merupakan sebuah method dengan nama `displayDetailAnjing`. Dapat dipanggil pada class lain.

“`{...}`” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [14], `super.display();`

“`super.display()`” berfungsi untuk memanggil method `display` dengan mengakses kelas induk.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [15], `System.out.println("Memiliki warna bulu : " + warnaBulu);`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Memiliki warna bulu : " + value `warnaBulu`.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [16], `System.out.print("Memiliki kemampuan :");`

“`System.out.print(...)`” berfungsi untuk mencetak angka atau karakter. Dalam baris ini mencetak "Memiliki kemampuan :”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [17-19], `for(String kemampuanAnjing : kemampuan) {...}`

Ini merupakan sebuah for-each yang berfungsi untuk mengiterasi setiap elemen di dalam atribut array `kemampuan`. `String` merupakan tipe data dari atribut `kemampuanAnjing` yang digunakan sebagai penyimpanan dari setiap elemen dalam array `kemampuan`.

“`{...}`” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [18], `System.out.print " " + kemampuanAnjing);`

“`System.out.print(...)`” berfungsi untuk mencetak angka atau karakter. Dalam baris ini mencetak " " + value `kemampuanAnjing`.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Main.java

Pada baris [1], `package soal2;`

“package soal2” berfungsi untuk mendeklarasikan package dengan nama soal2. Class Main berada dalam package soal2.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3], `import java.util.Scanner;`

Mengimpor class scanner yang terdapat di dalam package java.util. Berfungsi untuk membaca input dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [5-47], `public class Main`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama Main.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [6-46], `public static void main(String[] args)`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“static” berfungsi membuat suatu method tanpa perlu melakukan instansiasi terlebih dahulu.

“void” berfungsi untuk tidak mengembalikan nilai apapun.

“main” merupakan nama fungsi yang digunakan oleh java sebagai awal masuk ke program.

“String[] args” berfungsi sebagai parameter yang diperlukan oleh fungsi main. Parameter ini adalah array dari argument perintah yang bisa diteruskan ke program java.

Pada baris [7], `Scanner userInput = new Scanner(System.in);`

Berfungsi untuk membuat sebuah objek dari class Scanner atau dalam hal ini variabel userInput akan berisi object dari Scanner class dengan cara memanggil konstruktor Scanner(System.in). System in adalah input stream yang digunakan untuk menerima input dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [8], `String nama, ras, warnabulu, kemampuanInput;`

“String” merupakan tipe data dari variabel nama, ras, warnabulu, kemampuanInput.

“nama, ras, warnabulu, kemampuanInput” merupakan variabel dari sebuah class Main.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [10], `System.out.println("Pilih jenis hewan yang ingin diinputkan:");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Pilih jenis hewan yang ingin diinputkan:".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [11], `System.out.println("1 = Kucing");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "1 = Kucing:".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [12], `System.out.println("2 = Anjing");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "2 = Anjing:".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [13], `System.out.println("Masukan pilihan: ");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Masukan pilihan: ".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [14], `int pilihan = userInput.nextInt();`

“Int” merupakan tipe datanya, “pilihan” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextInt()`. <tipe data> <nama variabel> = <nilai variabel>.

“userInput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“nextInt()” berfungsi untuk membaca nilai Int dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [15], `userInput.nextLine();`

Baris ini berfungsi untuk membersihkan baris baru yang masih tersisa dalam buffer masukan. Baris ini diperlukan karena sebelumnya `nextInt()` hanya bisa membaca angka, tidak bisa membaca karakter baris baru yang mungkin tersisa dalam buffer setelahnya.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [17-45], `switch (pilihan) {...}`

“switch” merupakan percabangan dengan membandingkan nilai sebuah variabel dengan beberapa nilai. Jika proses perbandingan tersebut menghasilkan true, maka blok code akan di eksekusi. (pilihan) merupakan nama variabel.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [18], `case 1:`

Merupakan nilai yang diperiksa. Dalam hal ini jika nilai pilihan = 1 maka blok kode akan di eksekusi.

Pada baris [19], `System.out.print("Nama Hewan Peliharaan: ");`

“`System.out.print(...)`” berfungsi untuk mencetak angka atau karakter. Dalam baris ini mencetak " Nama Hewan Peliharaan: ”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [20], `nama = userInput.nextLine();`

“nama” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextLine()`. <nama variabel> = <nilai variabel>.

“`userInput`” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“`nextLine()`” berfungsi untuk membaca nilai String dari pengguna.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [21], `System.out.print("Ras: ");`

“`System.out.print(...)`” berfungsi untuk mencetak angka atau karakter. Dalam baris ini mencetak "Ras: ”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [22], `ras = userInput.nextLine();`

“ras” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextLine()`. <nama variabel> = <nilai variabel>.

“`userInput`” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“`nextLine()`” berfungsi untuk membaca nilai String dari pengguna.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [23], `System.out.print("Warna Bulu: ");`

“`System.out.print(...)`” berfungsi untuk mencetak angka atau karakter. Dalam baris ini mencetak "Warna Bulu: ”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [24], `warnabulu = userInput.nextLine();`

“warnabulu” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextLine()`. <nama variabel> = <nilai variabel>.

“`userInput`” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“`nextLine()`” berfungsi untuk membaca nilai String dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [25], `Kucing k = new Kucing(ras, nama, warnabulu);`

“Kucing” merupakan sebuah class.

“k” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class Kucing.

“new Kucing” membuat objek baru dari kelas Kucing.

“(ras, nama, warnabulu)” berfungsi untuk mengisi value atribut dari parameter yang sudah ditentukan. Pada baris ini mengisi value atribut dengan parameter r:value variabel ras, n: value dari variabel nama, w: value dari variabel warnabulu.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [26], `k.displayDetailKucing();`

“k.displayDetailKucing” berfungsi untuk memanggil method `displayDetailKucing` pada objek “k” dari class Kucing.

Pada baris [27], `break;`

Memberhentikan case 1.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [29], `case 2:`

Merupakan nilai yang diperiksa. Dalam hal ini jika nilai pilihan = 2 maka blok kode akan di eksekusi.

Pada baris [30], `System.out.print("Nama Hewan Peliharaan: ");`

“System.out.print(…)” berfungsi untuk mencetak angka atau karakter. Dalam baris ini mencetak " Nama Hewan Peliharaan: ".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [31], `nama = userInput.nextLine();`

“nama” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userinput.nextLine()`. <nama variabel> = <nilai variabel>.

“userinput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“nextLine()” berfungsi untuk membaca nilai String dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [32], `System.out.print("Ras: ");`

“System.out.print(…)” berfungsi untuk mencetak angka atau karakter. Dalam baris ini mencetak "Ras: ".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [33], `ras = userInput.nextLine();`

“ras” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextLine()`. <nama variabel> = <nilai variabel>.

“userInput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“`nextLine()`” berfungsi untuk membaca nilai String dari pengguna.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [34], `System.out.print("Warna Bulu: ");`

“`System.out.print(...)`” berfungsi untuk mencetak angka atau karakter. Dalam baris ini mencetak “Warna Bulu: ”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [35], `warnabulu = userInput.nextLine();`

“warnabulu” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextLine()`. <nama variabel> = <nilai variabel>.

“userInput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“`nextLine()`” berfungsi untuk membaca nilai String dari pengguna.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [36], `System.out.print("Kemampuan : ");`

“`System.out.print(...)`” berfungsi untuk mencetak angka atau karakter. Dalam baris ini mencetak “Kemampuan : ”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [37], `kemampuanInput = userInput.nextLine();`

“kemampuanInput” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextLine()`. <nama variabel> = <nilai variabel>.

“userInput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“`nextLine()`” berfungsi untuk membaca nilai String dari pengguna.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [38], `String[] kemampuan = kemampuanInput.split(", ");`

“String” merupakan tipe datanya “kemampuan” merupakan variabel yang diberi sebuah penamaan yang juga sebuah array, Lalu setelah itu adalah valuenya diambil dari value variabel `kemampuanInput`. <tipe data> <nama variabel> = <nilai variabel>.

“split(", ")” berfungsi untuk membagi value kemampuanInput menjadi beberapa bagian array bedarkan pemisah koma dan spasi (", ").

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [39], `Anjing a = new Anjing(nama, ras, warnabulu, kemampuan);`

“Anjing” merupakan sebuah class.

“a” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class Anjing.

“new Anjing” membuat objek baru dari kelas Anjing.

“(nama, ras, warnabulu, kemampuan)” berfungsi untuk mengisi value atribut dari parameter yang sudah ditentukan. Pada baris ini mengisi value atribut dengan parameter n:value variabel nama, r: value dari variabel ras, w: value dari variabel warnabulu., k: value dari variabel kemampuan.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [40], `a.displayDetailAnjing();`

“a.displayDetailAnjing” berfungsi untuk memanggil method displayDetailAnjing pada objek “a” dari class Anjing.

Pada baris [41], `break;`
Memberhentikan case 2.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [43], `default:`

Apabila tidak ada case yang sesuai maka default yang akan di eksekusi.

Pada baris [44], `System.out.println("Pilihan tidak valid");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Pilihan tidak valid".

“;” berfungsi untuk mengakhiri sebuah perintah.

D. Tautan GIT

<https://github.com/ryanmi04/Praktikum-Pemrograman-2-Paralel-1/tree/main/PRAKTIKUM%204/src/soal2>