

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN II**  
**MODUL 4**



**PEWARISAN/INHERITANCE & DIAGRAM KELAS**

**Oleh:**

**Rizki Adhitiya Maulana**

**NIM. 2410817110014**

**PROGRAM STUDI TEKNOLOGI INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS LAMBUNG MANGKURAT**  
**OKTOBER 2025**

**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 : Rizki Adhitiya Maulana  
NIM : 2410817110014

Menyetujui,  
Asisten Praktikum

Mengetahui,  
Dosen Penanggung Jawab Praktikum

Jovan Gilbert Natamasindah  
NIM. 2310817310002

Irham Maulani Abdul Gani, S.Kom.,  
M.Kom.  
NIP. 199710312025061009

## DAFTAR ISI

LEMBAR PENGESAHAN .....	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR .....	iv
DAFTAR TABEL .....	v
SOAL 1 .....	6
A. Pembahasan.....	7
SOAL 2 .....	9
A. Source Code .....	10
B. Output Program.....	11
C. Pembahasan.....	11
SOAL 3 .....	15
A. Source Code .....	17
B. Output Program.....	21
C. Pembahasan.....	21
TAUTAN GIT.....	29

## **DAFTAR GAMBAR**

Gambar 1. Struktur dasar tabel database transportasi .....	6
Gambar 2. Class Diagram Inheritance Soal No. 1 .....	8
Gambar 3. Screenshot Output Program Soal 2 .....	11
Gambar 4. Inheritance Hewan Peliharaan.....	15
Gambar 5. Screenshot Output Program Soal 3 .....	21

## **DAFTAR TABEL**

Tabel 1. Atribut Umum Soal No.1 .....	7
Tabel 2. Atribut Khusus Soal No.1 .....	7
Tabel 3. Struktur Kelas HewanPeliharaan Soal No.2 .....	9
Tabel 4. Input dan Output Soal No. 2 .....	9
Tabel 5. Source Code HewanPeliharaan.java Soal No. 2 .....	10
Tabel 6. Source Code Soal2Main.java Soal No. 2 .....	10
Tabel 7. Input dan Output Soal No. 3 .....	15
Tabel 8. Source Code HewanPeliharaan.java Soal No. 3 .....	17
Tabel 9. Source Code Cat.java Soal No.3 .....	17
Tabel 10. Source Code Dog.java Soal No.3.....	18
Tabel 11. Source Code Soal3Main.java Soal No. 3 .....	19

## SOAL 1

Bengkel Amay baru saja membuat sebuah sistem informasi berbasis online. Di dalam databasenya terdapat 3 table, yaitu: cars, ships, dan airplanes dengan bentuk sebagai berikut

cars	
id	integer
brand	varchar(100) NN
type	varchar(100)
production_year	integer
passenger_capacity	integer
transmission	varchar(50)
fuel	varchar(50)

ships	
id	integer
brand	varchar(100) NN
type	varchar(100)
production_year	integer
passenger_capacity	integer
maximum_weight	decimal(10,2)

airplanes	
id	integer
brand	varchar(100) NN
type	varchar(100)
production_year	integer
passenger_capacity	integer
airline	varchar(100)
maximum_altitude	integer
engine	varchar(50)

Gambar 1. Struktur dasar tabel database transportasi

Dari table-table database diatas, kamu diminta untuk:

- Melakukan abstraksi terhadap table-table tersebut dengan mencari atribut umum dan khusus di ketiganya. Jika sudah tulis di bagian pembahasan apa saja atribut hasil abstraksinya.
- Membuat class diagram dan lakukan inheritance (bentuknya kurang lebih seperti class diagram di nomor 2). Jadi nantinya akan terdapat 4 kotak didalam class diagram (Vehicle, Cars, Ships, dan Airplanes).

## A. Pembahasan

- Abstraksi Table Database (Cars, Ships, Airplanes)

Berikut ini hasil setelah dilakukannya abstraksi untuk ketiga tabel Cars, Ships, dan Airplanes. Dari proses tersebut, bisa dilihat kalau ada beberapa atribut yang bersifat umum dan ada juga yang bersifat khusus.

### ➤ Atribut Umum

Atribut umum adalah atribut yang dimiliki oleh ketiga tabel, adapun atributnya diantaranya:

*Tabel 1. Atribut Umum Soal No.1*

Atribut Umum	Tipe Data
Id	integer (Primary key)
Brand	varchar (100) NN
Type	varchar (100)
Production_year	integer
Passenger_capacity	Integer

### ➤ Atribut Khusus

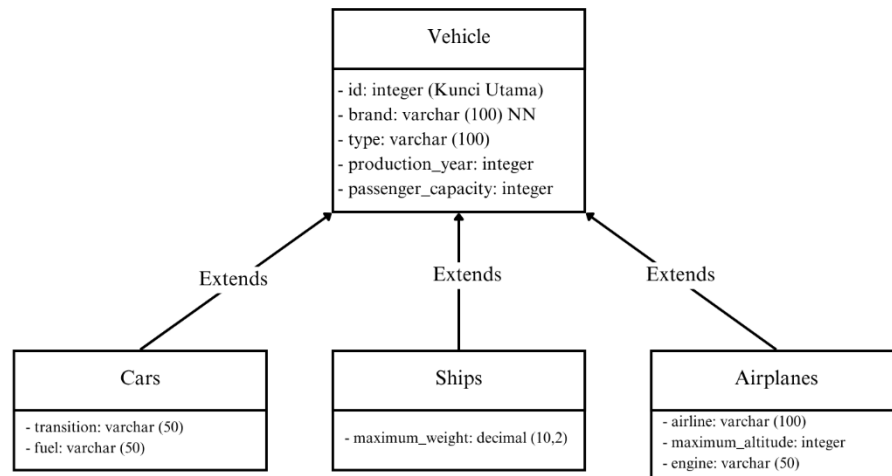
Atribut Khusus adalah atribut yang hanya dimiliki oleh masing-masing tabel, adapun atributnya diantaranya:

*Tabel 2. Atribut Khusus Soal No.1*

Tabel	Atribut Khusus	Tipe Data
Cars	Transition	varchar (50)
	Fuel	varchar (50)
Ships	Maximum_weight	Decima (10,2)
Airplanes	Airline	varchar (100)
	Maximum_altitude	integer
	Engine	varchar (50)

- Membuat Class Diagram (Inheritance)

Berdasarkan hasil abstraksi yang sudah dilakukan, kita akan membuat class diagram menggunakan konsep inheritance atau pewarisan. Yang mana nantinya akan terdapat 4 kotak didalam class diagram diantaranya Vehicle sebagai yang utama, dan Cars, Ships, serta Airplanes sebagai pewarisnya.



*Gambar 2. Class Diagram Inheritance Soal No. 1*



## SOAL 2

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

*Tabel 3. Struktur Kelas HewanPeliharaan Soal No.2*

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.

*Tabel 4. Input dan Output Soal No. 2*

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

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

## A. Source Code

*Tabel 5. Source Code HewanPeliharaan.java Soal No. 2*

1	package soall;
2	
3	public class HewanPeliharaan
4	{
5	private String name;
6	private String breed;
7	
8	public HewanPeliharaan(String name, String breed)
9	{
10	this.name = name;
11	this.breed = breed;
12	}
13	
14	public String getName()
15	{
16	return name;
17	}
18	
19	public String getBreed()
20	{
21	return breed;
22	}
23	
24	public void display()
25	{
26	System.out.println();
27	System.out.println("Detail Hewan Peliharaan:");
28	System.out.println("Nama hewan peliharaanku
29	adalah : " + getName());
30	System.out.println("Dengan ras : " +
31	getBreed());
	}
	}

*Tabel 6. Source Code Soal2Main.java Soal No. 2*

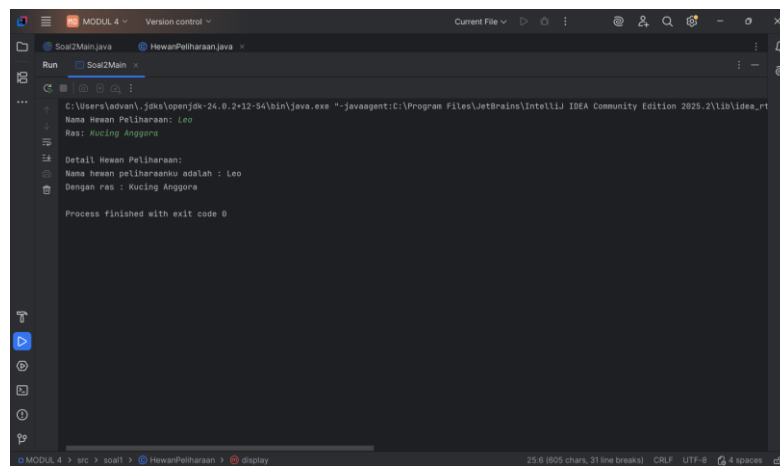
1	package soall;
2	
3	import java.util.Scanner;
4	
5	public class Soal2Main
6	{
7	public static void main(String[] args)
8	{
9	Scanner input = new Scanner(System.in);
10	

```

11      System.out.print("Nama Hewan Peliharaan: ");
12      String name = input.nextLine();
13
14      System.out.print("Ras: ");
15      String breed = input.nextLine();
16
17      HewanPeliharaan pet = new HewanPeliharaan(name,
18  breed);
19      pet.display();
20  }

```

## B. Output Program



Gambar 3. Screenshot Output Program Soal 2

## C. Pembahasan

- HewanPeliharaan.java

Pada baris [1] terdapat *package soal1*; dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal1.

Pada baris [3] terdapat *public class HewanPeliharaan* dimana berfungsi untuk membuat class publik yang mana nantinya semua kode program yang dibuat akan berada di dalam class ini.

Pada baris [5] terdapat *private String name*; dimana berfungsi untuk menyimpan nama untuk hewan dalam tipe data string, sehingga variabel yang ada akan menyimpan teks di dalamnya.

Pada baris [6] terdapat `private String breed;` dimana berfungsi untuk menyimpan ras untuk hewan dalam tipe data string, sehingga variabel yang ada akan menyimpan teks di dalamnya.

Pada baris [8] sampai [12] terdapat constructor `public HewanPeliharaan(String name, String breed)` dimana berfungsi untuk memberi nilai awal pada nama dan ras hewan ketika ingin membuat objek. `this.name = name;` dan `this.breed = breed;` akan menyimpan nilai yang dimasukkan ke dalam variabel name dan breed milik class HewanPeliharaan.

Pada baris [14] sampai [17] terdapat `public String getName()` dimana berfungsi untuk mengambil nilai dari variabel name. `return name;` akan mengembalikan isi dari variabel name agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [19] sampai [22] terdapat `public String getBreed()` dimana berfungsi untuk mengambil nilai dari variabel breed. `return breed;` akan mengembalikan isi dari variabel breed agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [24] terdapat `public void display()` dimana berfungsi untuk menampilkan seluruh informasi yang ingin ditampilkan ke layar.

Pada baris [26] sampai [29] terdapat `System.out.println` dimana berfungsi untuk menampilkan teks atau nilai yang ada ke layar. `getName` dan `getBreed` digunakan untuk menampilkan nilai yang ada di variabel ke layar.

Pada baris [4] sampai [31] terdapat `{ }` dimana digunakan untuk membatasi blok kode yang ada pada program.

- Soal2Main.java

Pada baris [1] terdapat `package soal1;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal1.

Pada baris [3] terdapat `import java.util.Scanner;` dimana berfungsi untuk mengimpor class Scanner yang berada di package java.util

sehingga program dapat membaca input yang dimasukkan oleh pengguna melalui keyboard.

Pada baris [5] terdapat `public class Soal2Main` dimana berfungsi untuk membuat class publik yang mana nantinya semua kode program yang dibuat akan berada di dalam class ini.

Pada baris [7] terdapat `public static void main(String[] args)` dimana berfungsi sebagai method main atau titik awal program, artinya program akan dijalankan mulai dari kode ini.

Pada baris [9] terdapat `Scanner input = new Scanner(System.in);` dimana berfungsi untuk menghadirkan objek Scanner dengan nama input untuk membaca atau mengambil input dari pengguna melalui keyboard dan memasukkannya ke dalam sistem (`System.in`).

Pada baris [11] terdapat `System.out.print("Nama Hewan Peliharaan: ");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [12] terdapat `String name = input.nextLine();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel `name`.

Pada baris [14] terdapat `System.out.print("Ras: ");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [15] terdapat `String breed = input.nextLine();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel `breed`.

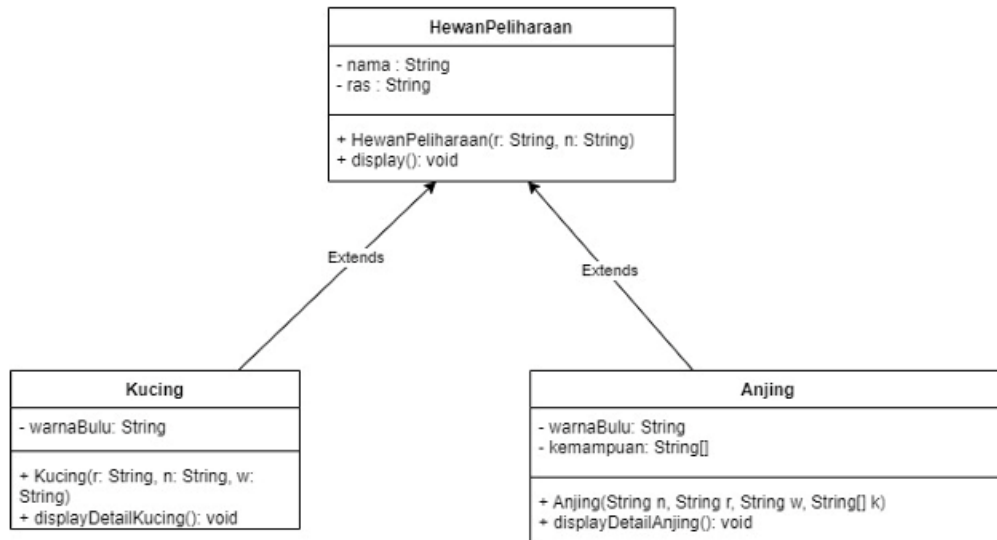
Pada baris [17] terdapat `HewanPeliharaan pet = new HewanPeliharaan(name, breed);` dimana berfungsi untuk membuat objek baru dari class `HewanPeliharaan` dengan mengirimkan nilai `name` dan `breed` sebagai parameter ke constructor.

Pada baris [18] terdapat `pet.display()`; dimana berfungsi untuk menampilkan seluruh informasi hewan peliharaan yang sudah dibuat sebelumnya pada method `display()` ke layar.

Pada baris [6] sampai [20] terdapat `{ }` dimana digunakan untuk membatasi blok kode yang ada pada program.

### SOAL 3

Diberikan class diagram seperti berikut:



Gambar 4. Inheritance Hewan Peliharaan

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 Input
- Isi detail dari class Kucing dan Anjing harus memanggil method display pada parent class terlebih dahulu.

Tabel 7. Input dan Output Soal No. 3

Input
Pilih jenis hewan yang ingin diinputkan: 1 = Kucing 2 = Anjing Masukkan pilihan: 1 Nama hewan peliharaan: lolo

Ras: Anggora Warna Bulu: abu-abu
<b>Output</b>
Detail Hewan Peliharaan: Nama hewan peliharaanku adalah : lolo Dengan ras : Anggora Memiliki warna bulu : abu-abu
<b>Input</b>
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
<b>Output</b>
Detail Hewan Peliharaan: Nama hewan peliharaanku adalah : andi Dengan ras : cihuahua Memiliki warna bulu : putih Memiliki kemampuan : berenang berjabat tangan melompat

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



## A. Source Code

*Tabel 8. Source Code HewanPeliharaan.java Soal No. 3*

1	package soal2;
2	
3	public class HewanPeliharaan
4	{
5	private String name;
6	private String breed;
7	
8	public HewanPeliharaan(String name, String breed)
9	{
10	this.name = name;
11	this.breed = breed;
12	}
13	
14	public String getName()
15	{
16	return name;
17	}
18	
19	public String getBreed()
20	{
21	return breed;
22	}
23	
24	public void display()
25	{
26	System.out.println();
27	System.out.println("Detail Hewan Peliharaan:");
28	System.out.println("Nama hewan peliharaanku
29	adalah : " + getName());
30	System.out.println("Dengan ras : " +
31	getBreed());
	}
	}

*Tabel 9. Source Code Cat.java Soal No.3*

1	package soal2;
2	
3	public class Cat extends HewanPeliharaan
4	{
5	private String furColor;
6	
7	public Cat(String name, String breed, String
8	furColor)
9	{
10	super(name, breed);

11	this.furColor = furColor;
12	}
13	
14	public String getFurColor()
15	{
16	return furColor;
17	}
18	
19	@Override
20	public void display()
21	{
22	super.display();
23	System.out.println("Memiliki warna bulu : " +
24	getFurColor());
25	}

*Tabel 10. Source Code Dog.java Soal No.3*

1	package soal2;
2	
3	public class Dog extends HewanPeliharaan
4	{
5	private String furColor;
6	private String[] abilityAnimal;
7	
8	public Dog(String name, String breed, String
9	furColor, String[] abilityAnimal)
10	{
11	super(name, breed);
12	this.furColor = furColor;
13	this.abilityAnimal = abilityAnimal;
14	}
15	public String getFurColor()
16	{
17	return furColor;
18	}
19	
20	public String[] getAbilityAnimal()
21	{
22	return abilityAnimal;
23	}
24	
25	@Override
26	public void display()
27	{
28	super.display();
29	System.out.println("Memiliki warna bulu : " +

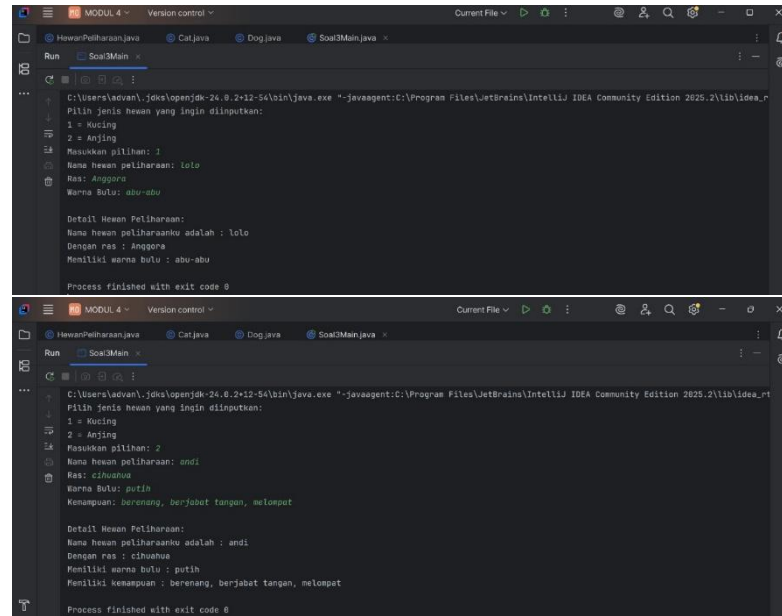
	getFurColor());
30	System.out.print("Memiliki kemampuan : ");
31	
32	for (int i = 0; i < getAbilityAnimal().length;
	i++)
33	{
34	System.out.print(getAbilityAnimal()[i]);
35	}
36	}
37	}

*Tabel 11. Source Code Soal3Main.java Soal No. 3*

1	package soal2;
2	
3	import java.util.Scanner;
4	
5	public class Soal3Main
6	{
7	public static void main(String[] args)
8	{
9	Scanner input = new Scanner(System.in);
10	
11	System.out.println("Pilih jenis hewan yang ingin diinputkan:");
12	System.out.println("1 = Kucing");
13	System.out.println("2 = Anjing");
14	System.out.print("Masukkan pilihan: ");
15	int choice = input.nextInt();
16	input.nextLine();
17	
18	System.out.print("Nama hewan peliharaan: ");
19	String name = input.nextLine();
20	
21	System.out.print("Ras: ");
22	String breed = input.nextLine();
23	
24	if (choice == 1)
25	{
26	System.out.print("Warna Bulu: ");
27	String furColor = input.nextLine();
28	
29	Cat mycat = new Cat(name, breed, furColor);
30	mycat.display();
31	}
32	
33	else if (choice == 2)
34	{
35	System.out.print("Warna Bulu: ");

```
36         String furColor = input.nextLine();
37
38         System.out.print("Kemampuan: ");
39         String abilityInput = input.nextLine();
40         String[] abilityAnimal =
abilityInput.split(",");
41
42         Dog myDog = new Dog(name, breed, furColor,
abilityAnimal);
43         myDog.display();
44     }
45
46     else
47     {
48         System.out.println("Invalid choice!");
49     }
50     input.close();
51 }
52 }
```

## B. Output Program



```
Run
C:\Users\adnan\doks\openjdk-24.8.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2\lib\idea_r...
Pilih jenis hewan yang ingin diinputkan:
1 = Kucing
2 = Anjing
Masukkan 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

Run
C:\Users\adnan\doks\openjdk-24.8.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2\lib\idea_r...
Pilih jenis hewan yang ingin diinputkan:
1 = Kucing
2 = Anjing
Masukkan pilihan: 2
Nama hewan peliharaan: andi
Ras: sihuanua
Warna bulu: putih
Kemampuan: berenang, berjabat tangan, melompat

Detail Hewan Peliharaan:
Nama hewan peliharaanku adalah : andi
Dengan ras : sihuanua
Memiliki warna bulu : putih
Memiliki kemampuan : berenang, berjabat tangan, melompat

Process finished with exit code 0
```

Gambar 5. Screenshot Output Program Soal 3

## C. Pembahasan

- HewanPeliharaan.java

Pada baris [1] terdapat `package soal2;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal2.

Pada baris [3] terdapat `public class HewanPeliharaan` dimana berfungsi untuk membuat class publik yang mana nantinya semua kode program yang dibuat akan berada di dalam class ini.

Pada baris [5] terdapat `private String name;` dimana berfungsi untuk menyimpan nama untuk hewan dalam tipe data string, sehingga variabel yang ada akan menyimpan teks di dalamnya.

Pada baris [6] terdapat `private String breed;` dimana berfungsi untuk menyimpan ras untuk hewan dalam tipe data string, sehingga variabel yang ada akan menyimpan teks di dalamnya.

Pada baris [8] sampai [12] terdapat `constructor public HewanPeliharaan(String name, String breed)` dimana berfungsi untuk memberi nilai awal pada nama dan ras hewan ketika ingin

membuat objek. *this.name = name;* dan *this.breed = breed;* akan menyimpan nilai yang dimasukkan ke dalam variabel *name* dan *breed* milik class *HewanPeliharaan*.

Pada baris [14] sampai [17] terdapat *public String getName()* dimana berfungsi untuk mengambil nilai dari variabel *name*. *return name;* akan mengembalikan isi dari variabel *name* agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [19] sampai [22] terdapat *public String getBreed()* dimana berfungsi untuk mengambil nilai dari variabel *breed*. *return breed;* akan mengembalikan isi dari variabel *breed* agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [24] terdapat *public void display()* dimana berfungsi untuk menampilkan seluruh informasi yang ingin ditampilkan ke layar.

Pada baris [26] sampai [29] terdapat *System.out.println* dimana berfungsi untuk menampilkan teks atau nilai yang ada ke layar. *getName* dan *getBreed* digunakan untuk menampilkan nilai yang ada di variabel ke layar.

Pada baris [4] sampai [31] terdapat *{ }* dimana digunakan untuk membatasi blok kode yang ada pada program.

- Cat.java

Pada baris [1] terdapat *package soal2;* dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama *soal2*.

Pada baris [3] terdapat *public class Cat extends HewanPeliharaan* dimana berfungsi untuk membuat class publik yang merupakan turunan atau mewarisi semua atribut dan method dari class utama atau induk yaitu *HewanPeliharaan*.

Pada baris [5] terdapat *private String furColor;* dimana berfungsi untuk menyimpan warna bulu untuk hewan dalam tipe data string, sehingga variabel yang ada akan menyimpan teks di dalamnya.

Pada baris [7] sampai [11] terdapat *constructor public Cat(String name, String breed, String furColor)* dimana berfungsi untuk memberi nilai awal pada nama dan ras hewan ketika ingin membuat objek. *super(name, breed);* akan memanggil constructor milik class utama atau induk agar nilai nama dan ras yang ada dapat tersimpan dengan baik. *this.furColor = furColor;* akan menyimpan nilai yang dimasukkan ke dalam variabel *furColor* milik class *Cat*.

Pada baris [13] sampai [16] terdapat *public String getFurColor()* dimana berfungsi untuk mengambil nilai dari variabel *furColor*. *return furColor;* akan mengembalikan isi dari variabel *furColor* agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [18] sampai [23] terdapat *@Override public void display()* dimana berfungsi untuk menampilkan seluruh informasi yang ingin ditampilkan ke layar, *@Override* yang ada akan menimpa method yang ada sudah ada di class utama atau induk. *super.display();* memanggil method milik class utama atau induk agar dapat menampilkan informasi yang ada di dalam method. *System.out.println* dimana berfungsi untuk menampilkan teks atau nilai yang ada ke layar. *getFurColor()* digunakan untuk menampilkan nilai yang ada di variabel ke layar.

Pada baris [4] sampai [24] terdapat *{ }* dimana digunakan untuk membatasi blok kode yang ada pada program.

- Dog.java

Pada baris [1] terdapat *package soal2;* dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama *soal2*.

Pada baris [3] terdapat *public class Dog extends HewanPeliharaan* dimana berfungsi untuk membuat class publik yang merupakan turunan atau mewarisi semua atribut dan method dari class utama atau induk yaitu *HewanPeliharaan*.

Pada baris [5] terdapat `private String furColor;` dimana berfungsi untuk menyimpan warna bulu untuk hewan dalam tipe data string, sehingga variabel yang ada akan menyimpan teks di dalamnya.

Pada baris [6] terdapat `private String[] abilityAnimal;` dimana berfungsi untuk menyimpan kemampuan untuk hewan dalam tipe data array string, sehingga variabel yang ada akan menyimpan teks di dalamnya.

Pada baris [8] sampai [13] terdapat `constructor public Dog(String name, String breed, String furColor, String[] abilityAnimal)` dimana berfungsi untuk memberi nilai awal pada nama dan ras hewan ketika ingin membuat objek. `super(name, breed);` akan memanggil constructor milik class utama atau induk agar nilai nama dan ras yang ada dapat tersimpan dengan baik. `this.furColor = furColor;` akan menyimpan nilai yang dimasukkan ke dalam variabel `furColor` milik class `Dog`. `this.abilityAnimal = abilityAnimal;` akan menyimpan nilai yang dimasukkan ke dalam variabel `abilityAnimal` milik class `Dog`.

Pada baris [15] sampai [18] terdapat `public String getFurColor()` dimana berfungsi untuk mengambil nilai dari variabel `furColor`. `return furColor;` akan mengembalikan isi dari variabel `furColor` agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [20] sampai [23] terdapat `public String[] getAbilityAnimal()` dimana berfungsi untuk mengambil nilai dari variabel `abilityAnimal`. `Return abilityAnimal;` akan mengembalikan isi dari variabel `abilityAnimal` agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [25] sampai [30] terdapat `@Override public void display()` dimana berfungsi untuk menampilkan seluruh informasi yang ingin ditampilkan ke layar. `@Override` yang ada akan menimpa method yang



ada sudah ada di class utama atau induk. `super.display()`; memanggil method milik class utama atau induk agar dapat menampilkan informasi yang ada di dalam method. `System.out.println` dimana berfungsi untuk menampilkan teks atau nilai yang ada ke layar. `getFurColor()` dan `getAbilityAnimal()` digunakan untuk menampilkan nilai yang ada di variabel ke layar.

Pada baris [32] sampai [35] terdapat `for (int i = 0; i < getAbilityAnimal().length; i++)` berfungsi untuk menampilkan semua kemampuan hewan yang tersimpan dalam array `abilityAnimal`. Perulangan dimulai dari indeks 0 dan berjalan hingga elemen terakhir array. `System.out.print(getAbilityAnimal()[i]);` akan menampilkan kemampuan ke-i tanpa berpindah ke baris baru, sehingga seluruh kemampuan ditampilkan secara berurutan dalam satu baris.

Pada baris [4] sampai [37] terdapat `{ }` dimana digunakan untuk membatasi blok kode yang ada pada program.

- Soal3Main.java

Pada baris [1] terdapat `package soal2;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama `soal2`.

Pada baris [3] terdapat `import java.util.Scanner;` dimana berfungsi untuk mengimpor class `Scanner` yang berada di package `java.util` sehingga program dapat membaca input yang dimasukkan oleh pengguna melalui keyboard.

Pada baris [5] terdapat `public class Soal3Main` dimana berfungsi untuk membuat class publik yang mana nantinya semua kode program yang dibuat akan berada di dalam class ini.

Pada baris [7] terdapat `public static void main(String[] args)` dimana berfungsi sebagai method `main` atau titik awal program, artinya program akan dijalankan mulai dari kode ini.

Pada baris [9] terdapat `Scanner input = new Scanner(System.in);` dimana berfungsi untuk menghadirkan objek Scanner dengan nama input untuk membaca atau mengambil input dari pengguna melalui keyboard dan memasukkannya ke dalam sistem (`System.in`).

Pada baris [11] terdapat `System.out.print("Pilih jenis hewan yang ingin diinputkan: ");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [12] terdapat `System.out.print("1 = Kucing");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [13] terdapat `System.out.print("2 = Anjing");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [14] terdapat `System.out.print("Masukkan pilihan: ");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [15] terdapat `int choice = input.nextLine();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel `choice`.

Pada baris [16] terdapat `input.nextLine();` dimana berfungsi untuk membaca seluruh baris input sebagai angka atau bilangan bulat.

Pada baris [18] terdapat `System.out.print("Nama hewan peliharaan: ");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [19] terdapat `String name = input.nextLine();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel `name`.

Pada baris [21] terdapat `System.out.print("Ras: ");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [22] terdapat `String breed = input.nextLine();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel `breed`.

Pada baris [24] terdapat `if (choice == 1)` dimana berfungsi untuk menentukan jenis hewan yang dipilih oleh pengguna berdasarkan nilai dari variabel `choice`.

Pada baris [26] terdapat `System.out.print("Warna Bulu: ");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [27] terdapat `String furColor = input.nextLine();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel `furColor`.

Pada baris [29] terdapat `Cat mycat = new Cat(name, breed, furColor);` dimana berfungsi untuk membuat objek baru dari class `Cat` dengan mengirimkan nilai `name`, `breed` dan `furColor` sebagai parameter ke constructor.

Pada baris [30] terdapat `myCat.display();` dimana berfungsi untuk menampilkan seluruh informasi hewan peliharaan yang sudah dibuat sebelumnya pada method `display()` ke layar.

Pada baris [33] terdapat `else if (choice == 2)` dimana berfungsi untuk menentukan jenis hewan yang dipilih oleh pengguna berdasarkan nilai dari variabel `choice`.

Pada baris [35] terdapat `System.out.print("Warna Bulu: ");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [36] terdapat `String furColor = input.nextLine();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel `furColor`.

Pada baris [38] terdapat `System.out.print("Kemampuan: ");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [39] terdapat `String abilityInput = input.nextLine();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel `ability`.

Pada baris [40] terdapat `String[] abilityAnimal = abilityInput.split(",");` dimana berfungsi untuk memisahkan teks input berdasarkan tanda koma (,) dan menyimpannya ke dalam array `abilityAnimal`, sehingga setiap kemampuan tersimpan sebagai elemen terpisah.

Pada baris [42] terdapat `Dog myDog = new Dog(name, breed, furColor, ability);` dimana berfungsi untuk membuat objek baru dari class `Dog` dengan mengirimkan nilai `name`, `breed`, `furColor` dan `ability` sebagai parameter ke constructor.

Pada baris [43] terdapat `myDog.display();` dimana berfungsi untuk menampilkan seluruh informasi hewan peliharaan yang sudah dibuat sebelumnya pada method `display()` ke layar.

Pada baris [46] sampai [49] terdapat `else` dimana berfungsi untuk menangani kondisi ketika pilihan yang dipilih oleh pengguna diluar pilihan yang ada dari variabel `choice`. `System.out.print("Invalid choice");` dimana berfungsi untuk menampilkan teks ke layar.

Pada baris [50] terdapat `input.close();` dimana berfungsi untuk menutup objek scanner apabila telah selesai digunakan dalam program.

Pada baris [4] sampai [52] terdapat `{ }` dimana digunakan untuk membatasi blok kode yang ada pada program.

## **TAUTAN GIT**

<https://github.com/Rizki-A-M/PRAKTIKUM-PEMROGRAMAN-II.git>