

Nama : Kelvianto Pratama Harum

NIM : 200210500016

MK : Pemrograman Lanjut

Dosen : Muhammad Fajar B, S.Pd., M.Cs.

PERTEMUAN V

POLYMORPHISM

A. Contoh 5.1

Source code:

Mahasiswa.java

```
23      @Override
24      public void siapaKamu() {
25          System.out.println("Saya Mahasiswa");
26      }
```

Dosen.java

```
23      @Override
24      public void siapaKamu() {
25          System.out.println("Saya Dosen");
26      }
```

Contoh51.java

```
3      public class Contoh51 {
4          public static void main(String[] args) {
5              Dosen dosen = new Dosen();
6              Mahasiswa mahasiswa = new Mahasiswa();
7              dosen.siapaKamu();
8              mahasiswa.siapaKamu();
9          }
10     }
```

Output:

```
> Task :app:run
Saya Dosen
Saya Mahasiswa

BUILD SUCCESSFUL in 653ms
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Ini adalah modifikasi project pada Praktikum 4.1; Pertemuan IV – Enkapsulasi.

Pada class **Mahasiswa** dan **Dosen** ditambahkan sebuah method **siapaKamu()** yang menimpa method **siapaKamu()** pada class **Manusia**.

Keyword **@Override** menandakan method setelah keyword ini adalah **overriding** dari method yang ada pada **super** class.

Method ini di-**override** untuk memberikan output yang berbeda, pada **super** class **Manusia** output yang diberikan adalah console output **“Saya manusia”** sedangkan pada **subclass Mahasiswa** output yang diberikan adalah **“Saya Mahasiswa”** dan pada **subclass Dosen** output yang diberikan adalah **“Saya Dosen”**.

Hal ini dibuktikan pada saat pemanggilan method **siapaKamu()** pada object **dosen** dan **mahasiswa** pada Main class (**Contoh51**), terlihat pada output console yang diberikan sesuai dengan **overriding** yang telah dilakukan.

B. Contoh 5.2a

Source code:

```
1 package com.example.Pertemuan5;
2
3 class binatang {
4     int umur = 12;
5     int get_umur () {
6         return umur;
7     }
8 }
9 public class Contoh52a extends binatang {
10     int umur = 20;
11     public static void main(String args[]) {
12         Contoh52a h = new Contoh52a();
13         System.out.println(h.umur);
14     }
15 }
```

Output:

```
> Task :app:run
20

BUILD SUCCESSFUL in 531ms
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Pada contoh ini dibuat sebuah class bernama **binatang** yang memiliki **field** dengan nama **umur** tipe data **integer** dengan nilai **12**, dan sebuah method dengan nama

get_umur yang pada intinya mengembalikan nilai yang ada pada **field umur**. Seluruh **member** ini menggunakan modifier **default**.

Kemudian dibuat sebuah main class (**Contoh52a**) yang diturunkan dari class **binatang**, pada class ini **field umur** dioverride dengan deklarasi ulang dengan nilai **20**. Lalu pada main method dibuat sebuah object dari class **Contoh52a** dengan nama **h**. Kemudian nilai **field umur** pada object **h** dioutput pada console, yang dapat dilihat bernilai **20**, sesuai dengan yang telah dideklarasikan. Pada dasarnya program akan mencari field/method dengan nama tersebut didalam class itu sendiri, lalu jika tidak ditemukan akan dipanggil dari **super** class object tersebut.

C. Contoh 5.2b

Source code:

```
1 package com.example.Pertemuan5;
2
3 class binatang2 {
4     int umur = 12;
5     int get_umur () {
6         return umur;
7     }
8 }
9 public class Contoh52b extends binatang2 {
10     int umur = 20;
11     int get_umur() {
12         return 25;
13     }
14     public static void main (String args[]) {
15         Contoh52b h = new Contoh52b();
16         System.out.println(h.get_umur());
17     }
18 }
```

Output:

```
> Task :app:run
25

BUILD SUCCESSFUL in 565ms
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Pada contoh ini dibuat sebuah class bernama **binatang2** yang memiliki **field** dengan nama **umur** tipe data **integer** dengan nilai **12**, dan sebuah method dengan nama **get_umur** yang pada intinya mengembalikan nilai yang ada pada **field umur**. Seluruh **member** ini menggunakan modifier **default**.

Kemudian dibuat sebuah main class (**Contoh52b**) yang diturunkan dari class **binatang2**, pada class ini **field umur** dioverride dengan deklarasi ulang dengan nilai **20**. Method **get_umur** juga di-override dengan mengubah perilaku method menjadi mengembalikan nilai literal **25**. Lalu pada main method dibuat sebuah object dari class **Contoh52b** dengan nama **h**. Kemudian nilai balik method **get_umur** pada object **h** dioutput pada console, yang dapat dilihat bernilai **25**, sesuai dengan yang telah dideklarasikan. Pada dasarnya program akan mencari field/method dengan nama tersebut didalam class itu sendiri, lalu jika tidak ditemukan akan dipanggil dari **super** class object tersebut.

D. Contoh 5.3

Source code:

```
1      package com.example.Pertemuan5;
2
3      class binatang3 {
4          int umur = 12;
5          int get_umur() {
6              return umur;
7          }
8      }
9
10     public class Contoh53 extends binatang3 {
11         @Override
12         int get_umur() {
13             return super.get_umur();
14         }
15         public static void main(String args[]) {
16             Contoh53 h = new Contoh53();
17             System.out.println(h.get_umur());
18         }
19     }
```

Output:

```
> Task :app:run
12

BUILD SUCCESSFUL in 560ms
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Pada contoh ini dibuat sebuah class bernama **binatang3** yang memiliki **field** dengan nama **umur** tipe data **integer** dengan nilai **12**, dan sebuah method dengan nama **get_umur** yang pada intinya mengembalikan nilai yang ada pada **field umur**. Seluruh **member** ini menggunakan modifier **default**.

Kemudian dibuat sebuah main class (**Contoh53**) yang diturunkan dari class **binatang3**, pada class ini method **get_umur** di-override dengan mengubah perilaku method menjadi mengembalikan nilai balik pemanggilan method **get_umur** yang ada pada **super** class. Lalu pada main method dibuat sebuah object dari class **Contoh53** dengan nama **h**. Kemudian nilai balik method **get_umur** pada object **h** dioutput pada console, yang dapat dilihat bernilai **12**, sesuai dengan yang telah dideklarasikan. Pada dasarnya program akan mencari field/method dengan nama tersebut didalam class itu sendiri, lalu jika tidak ditemukan akan dipanggil dari **super** class object tersebut.

E. Contoh 5.4

Source code:

```
1  package com.example.Pertemuan5;
2
3  public class Contoh54 {
4      public static void ulangiHalo() {
5          for(int i = 1; i <= 3; i++)
6              System.out.println("Halo");
7      }
8      public static void ulangiHalo(int berapakali) {
9          for(int i = 1; i <= berapakali; i++)
10             System.out.println("Halo overloading");
11      }
12     public static void main(String args[]) {
13         ulangiHalo();
14         System.out.println();
15
16         ulangiHalo(4);
17     }
18 }
```

Output:

```
> Task :app:run
Halo
Halo
Halo

Halo overloading
Halo overloading
Halo overloading
Halo overloading

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Pada contoh ini dibuat sebuah class dengan nama **Contoh54** yang berisikan:

- Method **ulangiHalo** tanpa menggunakan parameter

Method ini akan memberikan output “**Halo**” pada console sebanyak 3 kali, dengan for-loop.

- Method **ulangiHalo** dengan parameter variabel **berapakali** tipe data **integer**

Method ini adalah **overloading** dari method dengan nama yang sama, dikatakan overload karena memiliki nama yang sama namun cara pemanggilan yang berbeda, pada method sebelumnya dipanggil tanpa parameter, sedangkan method ini dipanggil dengan sebuah argumen variable **integer**.

Method ini akan memberikan output “**Halo overloading**” sebanyak nilai **berapakali** saat pemanggilan method.

Kemudian pada method **main** dipanggil method **ulangiHalo()** tanpa parameter dan terlihat pada output ditampilkan string “**Halo**” sebanyak 3 kali. Lalu, dilakukan pemanggilan method **ulangiHalo** dengan argumen nilai **4**, terlihat pada output ditampilkan string “**Halo overloading**” sebanyak **4** kali sesuai nilai argumen. Intinya, program akan memilih method yang sesuai untuk dilakukan eksekusi sesuai dengan argumen yang diberikan.

F. Praktikum 5.1

Source code:

```
1 package com.example.Pertemuan5;
2
3 class BentukWajah {
4     public String respons(){
5         return "Perhatikan reaksi wajah saya";
6     }
7 }
8 class Senyum extends BentukWajah {
9     public String respons(){
10        return "Senyum karena senang";
11    }
12 }
13 class Tertawa extends BentukWajah {
14     public String respons(){
15        return "Tertawa karena gembira";
16    }
17 }
18 class Marah extends BentukWajah {
19     public String respons(){
20        return "Kemarahan disebabkan bertengkar";
21    }
22 }
23 class Sedih extends BentukWajah {
24     public String respons(){
25        return "Kesedihan disebabkan kecewa";
26    }
27 }
```

```

28 public class Praktikum51 {
29     public static void main(String args[]){
30         BentukWajah objBentuk = new BentukWajah();
31         Senyum objSenyum = new Senyum();
32         Tertawa objTertawa = new Tertawa();
33         Marah objMarah = new Marah();
34         Sedih objSedih = new Sedih();
35
36         BentukWajah[] Bentuk = new BentukWajah[5];
37         Bentuk[0] = objBentuk;
38         Bentuk[1] = objSenyum;
39         Bentuk[2] = objTertawa;
40         Bentuk[3] = objMarah;
41         Bentuk[4] = objSedih;
42
43         System.out.println("bentuk[0] : " + Bentuk[0].respons());
44         System.out.println("bentuk[1] : " + Bentuk[1].respons());
45         System.out.println("bentuk[2] : " + Bentuk[2].respons());
46         System.out.println("bentuk[3] : " + Bentuk[3].respons());
47         System.out.println("bentuk[4] : " + Bentuk[4].respons());
48     }
49 }

```

Output:

```

> Task :app:run
bentuk[0] : Perhatikan reaksi wajah saya
bentuk[1] : Senyum karena senang
bentuk[2] : Tertawa karena gembira
bentuk[3] : Kemarahan disebabkan bertengkar
bentuk[4] : Kesedihan disebabkan kecewa

BUILD SUCCESSFUL in 589ms
2 actionable tasks: 1 executed, 1 up-to-date

```

Penjelasan:

Pada contoh ini dibuat sebuah base class yaitu **BentukWajah** dengan sebuah method **respons** yang menampilkan string “Perhatikan reaksi wajah saya” pada console.

Lalu dibuat empat (4) subclass turunan dari class **BentukWajah**, yaitu:

- Class **Senyum** yang meng-**override** method **respons** menjadi menampilkan string “**Senyum karena senang**” pada console.
- Class **Tertawa** yang meng-**override** method **respons** menjadi menampilkan string “**Tertawa karena gembira**” pada console.
- Class **Marah** yang meng-**override** method **respons** menjadi menampilkan string “**Kemarahan disebabkan bertengkar**” pada console.
- Class **Sedih** yang meng-**override** method **respons** menjadi menampilkan string “**Kesedihan disebabkan kecewa**” pada console.

Kemudian, pada method **main** masing-masing class dibuat menjadi object, yaitu:

- Class **BentukWajah** dengan nama object **objBentuk**.

- Class **Senyum** dengan nama object **objSenyum**.
- Class **Tertawa** dengan nama object **objTertawa**.
- Class **Marah** dengan nama object **objMarah**.
- Class **Sedih** dengan nama object **objSedih**.

Lalu dibuat sebuah array dengan dari class **BentukWajah** dengan nama **Bentuk** dengan jumlah member **5 member**. Member tersebut adalah object yang telah dibuat sebelumnya, **objBentuk**, **objSenyum**, **objTertawa**, **objMarah**, dan **objSedih**.

Yang kemudian dipanggil method **respons** pada masing-masing member array tersebut, dan dapat dilihat pada console **method respons** yang digunakan adalah method yang sama dideklarasikan pada classnya.

G. Tantangan

Source code:

```
1 package com.example.Pertemuan5;
2
3 class Animal {
4     private String nama, sifat;
5     private int ukuran;
6     public String getNama(){
7         return nama;
8     }
9     public String getSifat(){
10        return sifat;
11    }
12    public int getUkuran(){
13        return ukuran;
14    }
15    public void setNama(String nama){
16        this.nama = nama;
17    }
18    public void setSifat(String sifat){
19        this.sifat = sifat;
20    }
21    public void setUkuran(int ukuran){
22        this.ukuran = ukuran;
23    }
24 }
25 class Mamalia extends Animal{
26     private String jenisMamalia;
27     private boolean bisaJalan;
28     private int jumlahKaki;
29     public Mamalia(){
30         super.setNama("Mamalia");
31         super.setSifat("Menyusui, dan bertulang belakang");
32         super.setUkuran(40);
33     }
34     public Mamalia(String nama){
35         super.setNama(nama);
36         super.setSifat("Menyusui, dan bertulang belakang");
37         super.setUkuran(40);
38     }
39     public boolean getBisaJalan(){
40         return bisaJalan;
41     }
42     public String getJenisMamalia(){
43         return jenisMamalia;
44     }
45     public int getJumlahKaki(){
46         return jumlahKaki;
47     }
48     public void setBisaJalan(boolean bisaJalan){
49         this.bisaJalan = bisaJalan;
50     }
51     public void setJenisMamalia(String jenisMamalia){
52         this.jenisMamalia = jenisMamalia;
53     }
54     public void setJumlahKaki(int jumlahKaki){
55         this.jumlahKaki = jumlahKaki;
56     }
57 }
```

```

58 class Aves extends Animal{
59     private String jenisAves;
60     private boolean bisaTerbang;
61     public Aves() {
62         super.setNama("Aves");
63         super.setSifat("Memiliki bulu (bukan rambut)");
64         super.setUkuran(20);
65     }
66     public Aves(String nama, double ukuran){
67         super.setNama(nama);
68         super.setUkuran((int)ukuran);
69         super.setSifat("Memiliki bulu (bukan rambut)");
70     }
71     public String getJenisAves(){
72         return jenisAves;
73     }
74     public boolean getBisaTerbang(){
75         return bisaTerbang;
76     }
77     public void setJenisAves(String jenisAves){
78         this.jenisAves = jenisAves;
79     }
80     public void setBisaTerbang(boolean bisaTerbang){
81         this.bisaTerbang = bisaTerbang;
82     }
83 }

```

```

84 class Ayam extends Aves{
85     private String jenisAyam;
86     private boolean bisaDiadu;
87     public Ayam() {
88         super.setBisaTerbang(false);
89         super.setJenisAves("Unggas");
90         super.setNama("Ayam");
91         super.setSifat("Berbulu layaknya Aves, namun tidak bisa terbang");
92         super.setUkuran(10);
93     }
94     public Ayam(String nama, double ukuran){
95         super.setBisaTerbang(false);
96         super.setJenisAves("Unggas");
97         super.setNama(nama);
98         super.setSifat("Berbulu layaknya Aves, namun tidak bisa terbang");
99         super.setUkuran((int) ukuran);
100    }
101    public String getJenisAyam(){
102        return jenisAyam;
103    }
104    public boolean getBisaDiadu(){
105        return bisaDiadu;
106    }
107    public void setJenisAyam(String jenisAyam){
108        this.jenisAyam = jenisAyam;
109    }
110    public void setBisaDiadu(boolean bisaDiadu){
111        this.bisaDiadu = bisaDiadu;
112    }
113 }

```

```

114 class Merpati extends Aves{
115     public Merpati(){
116         super.setBisaTerbang(true);
117         super.setJenisAves("Columbiformes");
118         super.setNama("Merpati");
119         super.setSifat("Bisa terbang, dan punya kicauan merdu");
120         super.setUkuran(5);
121     }
122     public Merpati(String nama, double ukuran){
123         super.setBisaTerbang(true);
124         super.setJenisAves("Columbiformes");
125         super.setNama(nama);
126         super.setSifat("Bisa terbang, dan punya kicauan merdu");
127         super.setUkuran((int)ukuran);
128     }
129     public void setBisaTerbang(boolean bisaTerbang){
130         super.setBisaTerbang(bisaTerbang);
131     }
132     public boolean getBisaTerbang(){
133         return super.getBisaTerbang();
134     }
135 }

```

```

136 public class Tantangan {
137     public static void main(String args[]){
138         Animal animal = new Animal();
139         Mamalia mamal1 = new Mamalia();
140         Mamalia mamal2 = new Mamalia("Kucing");
141         Aves aves1 = new Aves();
142         Aves aves2 = new Aves("Merak", 15.0d);
143         Ayam ayam1 = new Ayam();
144         Ayam ayam2 = new Ayam("Kate", 7.5d);
145         Merpati merpatil = new Merpati();
146         Merpati merpati2 = new Merpati("Pigeon", 5.2d);
147
148         //Animal
149         animal.setNama("Anjing");
150         animal.setSifat("Menyusui, dan bertulang belakang");
151         animal.setUkuran(12);
152         System.out.println("Nama: " + animal.getNama());
153         System.out.println("Sifat: " + animal.getSifat());
154         System.out.println("Ukuran: " + animal.getUkuran());
155         //end
156         System.out.println();

```

```

158         //Mamalia 1
159         mamal1.setNama("Manusia");
160         mamal1.setSifat("Menyusui, dan bertulang belakang");
161         mamal1.setUkuran(60);
162         mamal1.setBisaJalan(true);
163         mamal1.setJenisMamalia("Homo sapiens");
164         mamal1.setJumlahKaki(2);
165         System.out.println("Nama: " + mamal1.getNama());
166         System.out.println("Sifat: " + mamal1.getSifat());
167         System.out.println("Ukuran: " + mamal1.getUkuran());
168         System.out.println("Bisa jalan?: " + mamal1.getBisaJalan());
169         System.out.println("Jenis Mamalia: " + mamal1.getJenisMamalia());
170         System.out.println("Jumlah Kaki: " + mamal1.getJumlahKaki());
171         //end
172         System.out.println();
173
174         //Mamalia 2
175         mamal2.setSifat("Menyusui, dan bertulang belakang");
176         mamal2.setUkuran(11);
177         mamal2.setBisaJalan(true);
178         mamal2.setJenisMamalia("Felis catus");
179         mamal2.setJumlahKaki(4);
180         System.out.println("Nama: " + mamal2.getNama());
181         System.out.println("Sifat: " + mamal2.getSifat());
182         System.out.println("Ukuran: " + mamal2.getUkuran());
183         System.out.println("Bisa jalan?: " + mamal2.getBisaJalan());
184         System.out.println("Jenis Mamalia: " + mamal2.getJenisMamalia());
185         System.out.println("Jumlah Kaki: " + mamal2.getJumlahKaki());
186         //end
187         System.out.println();

```

```

189 //Aves 1
190 aves1.setNama("Merak");
191 aves1.setSifat("Memiliki bulu yang cantik pada bagian ekornya");
192 aves1.setUkuran(15);
193 aves1.setBisaTerbang(true);
194 aves1.setJenisAves("Pavo muticus");
195 System.out.println("Nama: " + aves1.getNama());
196 System.out.println("Sifat: " + aves1.getSifat());
197 System.out.println("Ukuran: " + aves1.getUkuran());
198 System.out.println("Bisa terbang?: " + aves1.getBisaTerbang());
199 System.out.println("Jenis Aves: " + aves1.getJenisAves());
200 //end
201 System.out.println();
202
203 //Aves 2
204 aves2.setSifat("Memiliki bulu yang cantik pada bagian ekornya");
205 aves2.setUkuran(23);
206 aves2.setBisaTerbang(true);
207 aves2.setJenisAves("Pavo muticus");
208 System.out.println("Nama: " + aves2.getNama());
209 System.out.println("Sifat: " + aves2.getSifat());
210 System.out.println("Ukuran: " + aves2.getUkuran());
211 System.out.println("Bisa terbang?: " + aves2.getBisaTerbang());
212 System.out.println("Jenis Aves: " + aves2.getJenisAves());
213 //end
214 System.out.println();

```

```

216 //Ayam 1
217 ayam1.setNama("Ayam");
218 ayam1.setSifat("Punya visual yang indah dan sering digunakan untuk adu");
219 ayam1.setUkuran(10);
220 ayam1.setBisaTerbang(false);
221 ayam1.setJenisAves("Gallus gallus domesticus");
222 ayam1.setBisaDiadu(true);
223 ayam1.setJenisAyam("Bangkok");
224 System.out.println("Nama: " + ayam1.getNama());
225 System.out.println("Sifat: " + ayam1.getSifat());
226 System.out.println("Ukuran: " + ayam1.getUkuran());
227 System.out.println("Bisa terbang?: " + ayam1.getBisaTerbang());
228 System.out.println("Jenis Aves: " + ayam1.getJenisAves());
229 System.out.println("Bisa diadu?: " + ayam1.getBisaDiadu());
230 System.out.println("Jenis Ayam: " + ayam1.getJenisAyam());
231 //end
232 System.out.println();
233
234 //Ayam 2
235 ayam2.setSifat("Punya visual yang indah dan sering digunakan untuk adu");
236 ayam2.setBisaTerbang(false);
237 ayam2.setJenisAves("Gallus gallus domesticus");
238 ayam2.setBisaDiadu(true);
239 ayam2.setJenisAyam("Kate");
240 System.out.println("Nama: " + ayam2.getNama());
241 System.out.println("Sifat: " + ayam2.getSifat());
242 System.out.println("Ukuran: " + ayam2.getUkuran());
243 System.out.println("Bisa terbang?: " + ayam2.getBisaTerbang());
244 System.out.println("Jenis Aves: " + ayam2.getJenisAves());
245 System.out.println("Bisa diadu?: " + ayam2.getBisaDiadu());
246 System.out.println("Jenis Ayam: " + ayam2.getJenisAyam());
247 //end
248 System.out.println();

```

```

249
250 //Merpati 1
251 merpatil.setNama("Pigeon");
252 merpatil.setSifat("Tubuh yang kecil, sering ditemukan di jalan");
253 merpatil.setUkuran(5);
254 merpatil.setBisaTerbang(true);
255 merpatil.setJenisAves("Columba livia");
256 System.out.println("Nama: " + merpatil.getNama());
257 System.out.println("Sifat: " + merpatil.getSifat());
258 System.out.println("Ukuran: " + merpatil.getUkuran());
259 System.out.println("Bisa terbang?: " + merpatil.getBisaTerbang());
260 System.out.println("Jenis Aves: " + merpatil.getJenisAves());
261 //end
262 System.out.println();
263
264 //Merpati 2
265 merpati2.setSifat("Bisa terbang, dan punya kicauan merdu");
266 merpati2.setBisaTerbang(true);
267 merpati2.setJenisAves("Columba livia");
268 System.out.println("Nama: " + merpati2.getNama());
269 System.out.println("Sifat: " + merpati2.getSifat());
270 System.out.println("Ukuran: " + merpati2.getUkuran());
271 System.out.println("Bisa terbang?: " + merpati2.getBisaTerbang());
272 System.out.println("Jenis Aves: " + merpati2.getJenisAves());
273 //end
274 }
275 }
276
277

```

Output:

```

> Task :app:run
Nama: Anjing
Sifat: Menyusui, dan bertulang belakang
Ukuran: 12

Nama: Manusia
Sifat: Menyusui, dan bertulang belakang
Ukuran: 60
Bisa jalan?: true
Jenis Mamalia: Homo sapiens
Jumlah Kaki: 2

Nama: Kucing
Sifat: Menyusui, dan bertulang belakang
Ukuran: 11
Bisa jalan?: true
Jenis Mamalia: Felis catus
Jumlah Kaki: 4

Nama: Merak
Sifat: Memiliki bulu yang cantik pada bagian ekornya
Ukuran: 15
Bisa terbang?: true
Jenis Aves: Pavo muticus

Nama: Merak
Sifat: Memiliki bulu yang cantik pada bagian ekornya
Ukuran: 23
Bisa terbang?: true
Jenis Aves: Pavo muticus

Nama: Ayam
Sifat: Punya visual yang indah dan sering digunakan untuk adu
Ukuran: 10
Bisa terbang?: false
Jenis Aves: Gallus gallus domesticus
Bisa diadu?: true
Jenis Ayam: Bangkok

```

```

Nama: Kate
Sifat: Punya visual yang indah dan sering digunakan untuk adu
Ukuran: 7
Bisa terbang?: false
Jenis Aves: Gallus gallus domesticus
Bisa diadu?: true
Jenis Ayam: Kate

Nama: Pigeon
Sifat: Tubuh yang kecil, sering ditemukan di jalan
Ukuran: 5
Bisa terbang?: true
Jenis Aves: Columba livia

Nama: Pigeon
Sifat: Bisa terbang, dan punya kicauan merdu
Ukuran: 5
Bisa terbang?: true
Jenis Aves: Columba livia

BUILD SUCCESSFUL in 646ms
2 actionable tasks: 1 executed, 1 up-to-date

```

Penjelasan:

1. ``package com.example.Pertemuan5;``: Mendefinisikan paket tempat kelas-kelas tersebut berada.
2. ``class Animal {``: Mendeklarasikan kelas ``Animal``.
3. ``private String nama, sifat;``: Mendefinisikan dua atribut ``String`` privat: ``nama`` dan ``sifat``.
4. ``private int ukuran;``: Mendefinisikan satu atribut ``int`` privat: ``ukuran``.
- 5-13. Metode getter dan setter untuk atribut ``nama``, ``sifat``, dan ``ukuran``.
- 14-49. Kelas ``Mamalia`` yang merupakan subkelas dari ``Animal``:
 - ``private String jenisMamalia;``: Atribut tambahan ``jenisMamalia``.
 - ``private boolean bisaJalan;``: Atribut tambahan ``bisaJalan``.
 - ``private int jumlahKaki;``: Atribut tambahan ``jumlahKaki``.
 - Konstruktor default yang mengatur nilai awal atribut ``nama``, ``sifat``, dan ``ukuran`` menggunakan metode setter dari kelas induk.
 - Konstruktor yang menerima parameter ``String nama``.
 - Metode getter dan setter untuk atribut tambahan ``jenisMamalia``, ``bisaJalan``, dan ``jumlahKaki``.
- 50-80. Kelas ``Aves`` yang merupakan subkelas dari ``Animal``:
 - ``private String jenisAves;``: Atribut tambahan ``jenisAves``.

- ``private boolean bisaTerbang;``: Atribut tambahan ``bisaTerbang``.
- Konstruktor default yang mengatur nilai awal atribut ``nama``, ``sifat``, dan ``ukuran`` menggunakan metode setter dari kelas induk.
- Konstruktor yang menerima parameter ``String nama`` dan ``double ukuran``.
- Metode getter dan setter untuk atribut tambahan ``jenisAves`` dan ``bisaTerbang``.

81-109. Kelas ``Ayam`` yang merupakan subkelas dari ``Aves``:

- ``private String jenisAyam;``: Atribut tambahan ``jenisAyam``.
- ``private boolean bisaDiadu;``: Atribut tambahan ``bisaDiadu``.
- Konstruktor default yang mengatur nilai awal atribut ``nama``, ``sifat``, dan ``ukuran`` menggunakan metode setter dari kelas induk.
- Konstruktor yang menerima parameter ``String nama`` dan ``double ukuran``.
- Metode getter dan setter untuk atribut tambahan ``jenisAyam`` dan ``bisaDiadu``.

110-125. Kelas ``Merpati`` yang merupakan subkelas dari ``Aves``:

- Konstruktor default yang mengatur nilai awal atribut ``nama``, ``sifat``, dan ``ukuran`` menggunakan metode setter dari kelas induk.
- Konstruktor yang menerima parameter ``String nama`` dan ``double ukuran``.
- Metode getter dan setter untuk atribut tambahan ``bisaTerbang``.

126-238. Kelas ``Tantangan`` dengan metode ``main``:

- Membuat beberapa objek dari kelas ``Animal``, ``Mamalia``, ``Aves``, ``Ayam``, dan ``Merpati``.
 - Mengatur nilai atribut objek-objek tersebut menggunakan metode setter.
 - Menampilkan nilai atribut objek-objek tersebut menggunakan metode getter
- Metode ``main`` mengilustrasikan penggunaan kelas dan hierarki pewarisan dalam kode dengan membuat objek-objek dan memanipulasi atribut-atributnya, kemudian menampilkan hasilnya ke konsol.