

LAPORAN PEMROGRAMAN BERORIENTASI OBJEK

“PRAKTIKUM 6”



Oleh :

Aisyah Aqilah Rian Vania

21091397002 (2021-B)

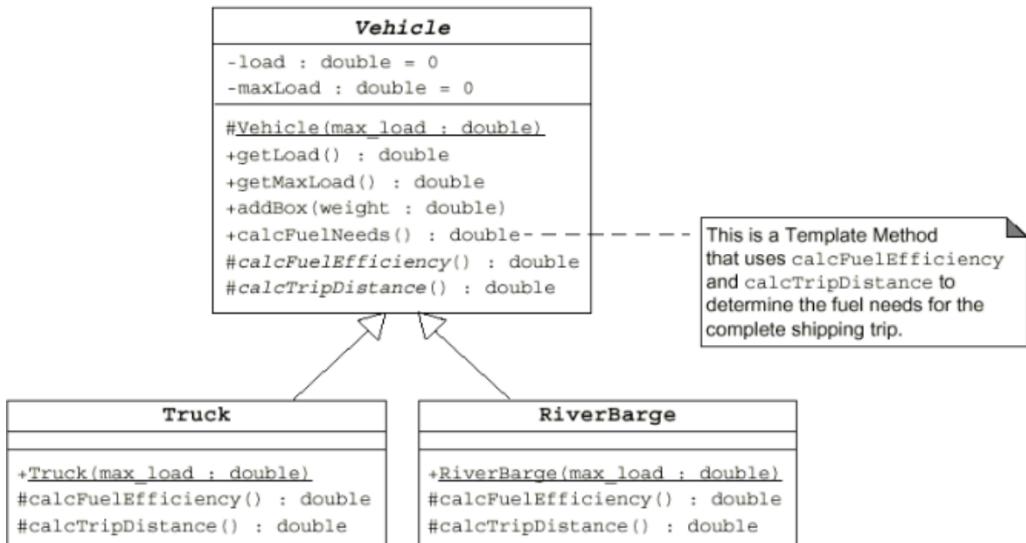
PROGRAM STUDI D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

2022

- Buat program berdasarkan UML berikut



Pemrograman :

```

public class prak6_no1 {
    public static void main(String[] args) {
        Truck truck_1 = new Truck();
        RiverBarge riverBarge_1 = new RiverBarge();

        System.out.println("== Truck ==");
        truck_1.getLoad(20);
        truck_1.getMaxLoad(max_load: 200);
        truck_1.addBox( weight: 20);
        truck_1.calcFuelEfficiency(150);
        truck_1.calcTripDistance(30000);
        System.out.println("\n");

        System.out.println("== River Barge ==");
        riverBarge_1.getLoad(30);
        riverBarge_1.getMaxLoad(max_load: 500);
        riverBarge_1.addBox( weight: 50.5);
        riverBarge_1.calcFuelEfficiency(300);
        riverBarge_1.calcTripDistance(70000);
        System.out.println("\n");
    }
}
  
```

The screenshot shows an IDE interface with the project **pbo_prak6** open. The **prak6_no1.java** file is selected in the editor. The code implements the **Vehicle** template method pattern. It creates instances of **Truck** and **RiverBarge** and demonstrates their behavior by setting load, calculating fuel efficiency, and calculating trip distance.

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help pbo_prak6 - prak6_no1.java

Project pbo_prak6 C:\Users\asus\IdeaProjects\pbo\pbo.prak6

src prak6_no1.java prak6_no2.java prak6_no3.java prak6_no4.java

prak6_no1.java

```
2 usages 2 inheritors
25 class Vehicle{
26     double load = 0;
27     double max_load = 0;
28
29     2 usages
30     void getLoad(double load) {
31         System.out.println("Load : " + load);
32     }
33     2 usages
34     void getMaxLoad(double max_load) {
35         System.out.println("Max Load : " + max_load);
36     }
37
38     2 usages
39     void addBox(double weight) {
40         System.out.println("Add Box : " + weight);
41     }
42
43     2 usages 2 overrides
44     public void calcFuelEfficiency(double fuel) {
45         System.out.println("Fuel Efficiency : " + fuel);
46     }
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
```

Run: prak6_no4

Version Control Run TODO Problems Terminal Services Build

Build completed successfully in 2 sec, 734 ms (6 minutes ago)

11:32 LF UTF-8 4 spaces 19.21 20/10/2022

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help pbo_prak6 - prak6_no1.java

Project pbo_prak6 C:\Users\asus\IdeaProjects\pbo\pbo.prak6

src prak6_no1.java prak6_no2.java prak6_no3.java prak6_no4.java

prak6_no1.java

```
2 usages 2 overrides
44 void calcTripDistance(double trip) {
45     System.out.println("Trip Distance : " + trip);
46 }
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
```

Run: prak6_no4

Version Control Run TODO Problems Terminal Services Build

Build completed successfully in 2 sec, 734 ms (6 minutes ago)

11:32 LF UTF-8 4 spaces 19.21 20/10/2022

```
prak6_no1.java
2 usages
54 void calcTripDistance(double trip) {
55     System.out.println("Trip Distance : " + trip);
56 }
2 usages
59 class RiverBarge extends Vehicle {
60     2 usages
61     public void calcFuelEfficiency(double fuel) {
62         System.out.println("Fuel Efficiency : " + fuel);
63     }
64     2 usages
65     void calcTripDistance(double trip) {
66         System.out.println("Trip Distance : " + trip);
67     }
}

Run: prak6_no4
Type here to search
```

Hasil :

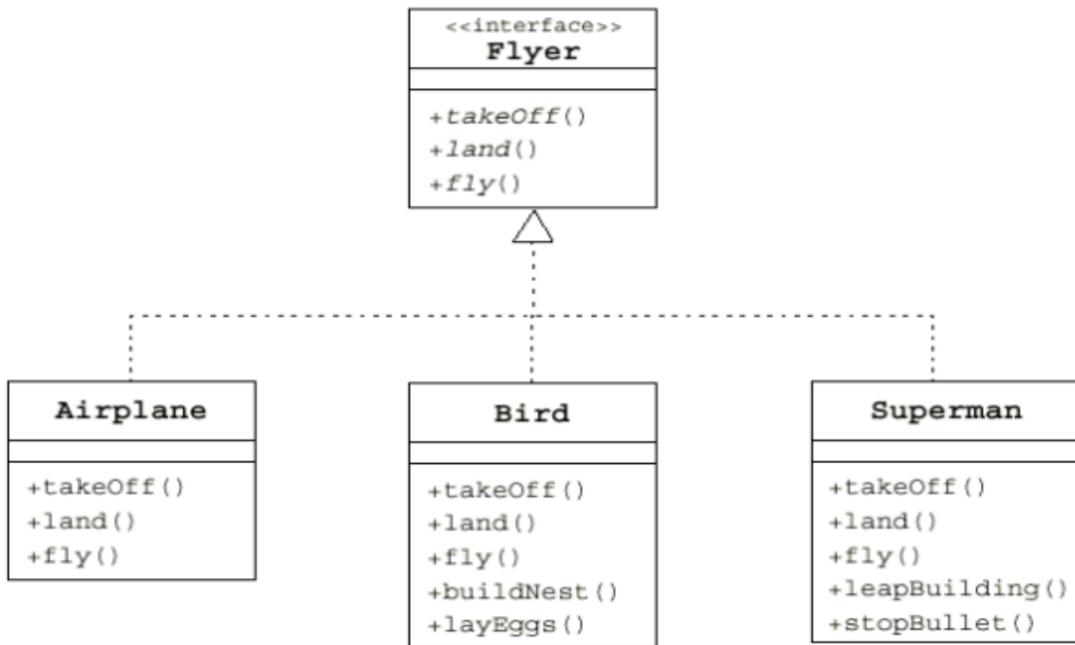
```
"C:\Program Files\Eclipse Adoptium\jdk-17.0.4.101-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8 prak6_no1
== Truck ==
Load : 20.0
Max Load : 200.0
Add Box : 20.0
Fuel Efficiency : 150.0
Trip Distance : 30000.0

== River Barge ==
Load : 50.0
Max Load : 500.0
Add Box : 50.5
Fuel Efficiency : 300.0
Trip Distance : 70000.0

Process finished with exit code 0
```

Analisa : Dari kode program diatas terdapat 3 class yaitu vehicle, truck, riverbarge dimana class vehicle nantinya akan mewariskan metode nya ke class child nya. Didalam class truck dan riverbarge terdapat 2 metode yang sama yaitu metode bensin dan metode yang mengetahui jarak tempuh. Kemudian metode truck dan riverbarge dipanggil difungsi main.

2. Buat program berdasarkan UML berikut



Pemrograman :

```
public class prak6_no2 {
    public static void main(String[] args) {
        System.out.println("== Pesawat ==");
        Flyer airplane = new Airplane();
        System.out.println(airplane.takeOff());
        System.out.println(airplane.land());
        System.out.println(airplane.fly());

        System.out.println("\n== Burung ==");
        Flyer bird = new Bird();
        System.out.println(bird.takeOff());
        System.out.println(bird.land());
        System.out.println(bird.fly());
        System.out.println(((Bird) bird).buildNest());
        System.out.println(((Bird) bird).layEggs());

        System.out.println("\n== Superman ==");
        Flyer superman = new Superman();
        System.out.println(superman.takeOff());
        System.out.println(superman.land());
        System.out.println(superman.fly());
        System.out.println(((Superman) superman).leapBuilding());
        System.out.println(((Superman) superman).stopBullet());
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the project 'pbo_prak6' open. The 'prak6_no2.java' file is the active editor, displaying the Java code that implements the Flyer interface for three different classes: Airplane, Bird, and Superman. The code uses System.out.println statements to demonstrate the behavior of each class's methods. The code is well-structured with clear comments and variable names.

The screenshot shows the IntelliJ IDEA interface with the project 'pbo_prak6' open. The code editor displays the 'prak6_no4.java' file, which contains the following Java code:

```
interface Flyer{
    String takeoff();
    String land();
    String fly();
}

class Airplane implements Flyer {
    @Override
    public String takeoff() {
        return "+ Pesawat Lepas Landas Dari Bandara Soekarno Hatta Ke Bandara Incheon";
    }

    @Override
    public String land() {
        return "+ Pesawat Berhasil Mendarat Di Bandara Incheon";
    }
}
```

The code editor shows code completion and navigation hints. The status bar at the bottom indicates the build completed successfully in 2 sec, 734 ms (8 minutes ago) at 17:1 CRLF UTF-8 4 spaces.

The screenshot shows the IntelliJ IDEA interface with the project 'pbo_prak6' open. The code editor displays the 'prak6_no4.java' file, which contains the following Java code:

```
@Override
public String fly() {
    return "+ Pesawat Terbang Selama 7 Jam";
}

class Bird implements Flyer{
    @Override
    public String takeoff() {
        return "+ Burung Kutilang Terbang Dari Ketinggian 900 Meter Diatas Permukaan Laut";
    }

    @Override
    public String land() {
        return "+ Burung Kutilang Mendarat Tepat Pada Atap Rumah Pak Andi";
    }
}
```

The code editor shows code completion and navigation hints. The status bar at the bottom indicates the build completed successfully in 2 sec, 734 ms (8 minutes ago) at 17:1 CRLF UTF-8 4 spaces.

The screenshot shows the IntelliJ IDEA interface with the project 'pbo_prak6' open. The 'src' directory contains four files: prak6_no1.java, prak6_no2.java, prak6_no3.java, and prak6_no4.java. The current file is prak6_no2.java. The code defines a class 'Burung Kutilang' that implements the 'Flyer' interface. It has three overridden methods: fly(), buildNest(), and layEggs(). The code also includes a class 'Superman' that implements the 'Flyer' interface, with methods takeOff(), land(), fly(), and leapBuilding(). The code is annotated with JavaDoc comments describing the actions performed by each method.

```
prak6_no1.java x prak6_no2.java x prak6_no3.java x prak6_no4.java x
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help pbo_prak6 - prak6_no2.java
Project pbo_prak6 C:\Users\asus\IdeaProjects\pbo\pbo.prak6
  > .idea
  > out
  > src
    prak6_no1.java
    prak6_no2.java
    prak6_no3.java
    prak6_no4.java
    pbo_prak6.iml
  External Libraries
  Scratches and Consoles
  Notifications
  Structure
  Bookmarks
Run: prak6_no4 x
Version Control Run TODO Problems Terminal Services Build
Build completed successfully in 2 sec, 734 ms (8 minutes ago)
17:1 CRLF UTF-8 4 spaces 19.23 20/10/2022
Type here to search
```

```
3 usages
@Override
public String fly() {
    return "+ Burung Kutilang Terbang Selama 45 Menit";
}

1 usage
@Override
public String buildNest() {
    return "+ Burung Kutilang Membangun Sarangnya Di Batang Pohon Yang Tertutup";
}

1 usage
@Override
public String layEggs() {
    return "+ Burung Kutilang Menggerami Telurnya";
}

3 usages
class Superman implements Flyer {

    3 usages
    @Override
    public String takeOff() {
        return "+ Superman Terbang Dengan Cepat Menuju Tempat Terjadinya Pembunuhan";
    }

    3 usages
    @Override
    public String land() {
        return "+ Superman Mendarat Di Tempat Terjadinya Pembunuhan";
    }

    3 usages
    @Override
    public String fly() {
        return "+ Superman Terbang Dengan Kecepatan 80 Km/jam";
    }

    1 usage
    public String leapBuilding() {
        return "+ Superman Terbang Melewati Banyak Gedung Tinggi";
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the project 'pbo_prak6' open. The 'src' directory contains four files: prak6_no1.java, prak6_no2.java, prak6_no3.java, and prak6_no4.java. The current file is prak6_no2.java. The code defines a class 'Superman' that implements the 'Flyer' interface, with methods takeOff(), land(), fly(), and leapBuilding(). The code is annotated with JavaDoc comments describing the actions performed by each method.

```
prak6_no1.java x prak6_no2.java x prak6_no3.java x prak6_no4.java x
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help pbo_prak6 - prak6_no2.java
Project pbo_prak6 C:\Users\asus\IdeaProjects\pbo\pbo.prak6
  > .idea
  > out
  > src
    prak6_no1.java
    prak6_no2.java
    prak6_no3.java
    prak6_no4.java
    pbo_prak6.iml
  External Libraries
  Scratches and Consoles
  Notifications
  Structure
  Bookmarks
Run: prak6_no4 x
Version Control Run TODO Problems Terminal Services Build
Build completed successfully in 2 sec, 734 ms (8 minutes ago)
17:1 CRLF UTF-8 4 spaces 19.23 20/10/2022
Type here to search
```

```
3 usages
class Superman implements Flyer {

    3 usages
    @Override
    public String takeOff() {
        return "+ Superman Terbang Dengan Cepat Menuju Tempat Terjadinya Pembunuhan";
    }

    3 usages
    @Override
    public String land() {
        return "+ Superman Mendarat Di Tempat Terjadinya Pembunuhan";
    }

    3 usages
    @Override
    public String fly() {
        return "+ Superman Terbang Dengan Kecepatan 80 Km/jam";
    }

    1 usage
    public String leapBuilding() {
        return "+ Superman Terbang Melewati Banyak Gedung Tinggi";
    }
}
```

```
prak6_no1.java:
88     return "+ Superman Mendarat Di Tempat Terjadinya Pembunuhan";
89 }
90
91     3 usages
92     @Override
93     public String fly() {
94         return "+ Superman Terbang Dengan Kecepatan 80 Km/jam";
95     }
96
97     1 usage
98     public String leapBuilding() {
99         return "+ Superman Terbang Melewati Banyak Gedung Tinggi";
100    }
101
102    1 usage
103    public String stopBullet() {
104        return "+ Superman Tidak Bisa Ditembak Peluru";
105    }
106}
```

Run: prak6_no4

Version Control Run TODO Problems Terminal Services Build

Build completed successfully in 2 sec. 734 ms (8 minutes ago)

17:1 CRLF UTF-8 4 spaces 19.23 20/10/2022

Hasil :

```
prak6_no2.java
prak6_no2.java
prak6_no1.java
prak6_no2.java
prak6_no3.java

Run: prak6_no2
C:\Program Files\Eclipse Adoptium\jdk-17.0.4.101-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8
+ Pesawat ===
+ Pesawat Lepas Landas Dari Bandara Soekarno Hatta Ke Bandara Incheon
+ Pesawat Berhasil Mendarat Di Bandara Incheon
+ Pesawat Terbang Selama 7 Jam

+ Burung ===
+ Burung Kutilang Terbang Dari Ketinggian 900 Meter Diatas Permukaan Laut
+ Burung Kutilang Mendarat Tepat Pada Atap Rumah Pak Andi
+ Burung Kutilang Terbang Selama 45 Menit
+ Burung Kutilang Membangun Sarangnya Di Batang Pohon Yang Tertutup
+ Burung Kutilang Menggerami Telurnya

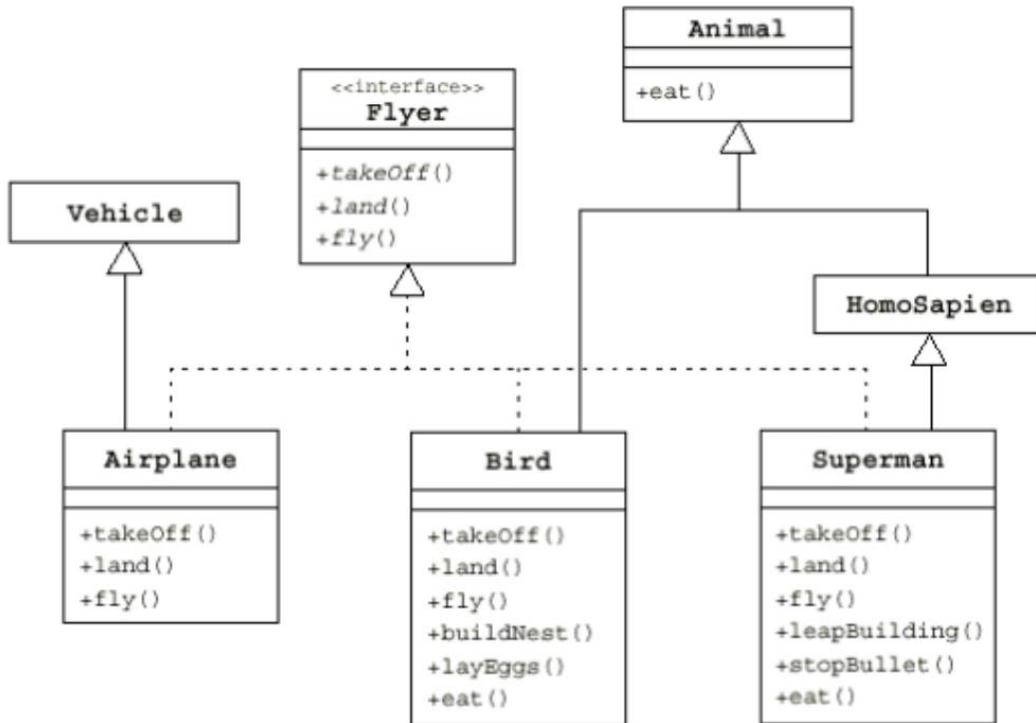
+ Superman ===
+ Superman Terbang Dengan Cepat Menuju Tempat Terjadinya Pembunuhan
+ Superman Mendarat Di Tempat Terjadinya Pembunuhan
+ Superman Terbang Dengan Kecepatan 80 Km/jam
+ Superman Terbang Melewati Banyak Gedung Tinggi
+ Superman Tidak Bisa Ditembak Peluru

Process finished with exit code 0

Build completed successfully in 2 sec. 684 ms (moments ago)
22:1 CRLF UTF-8 4 spaces 18.59 20/10/2022
```

Analisa : Pada kode program diatas terdapat satu interface yang bernama flyer. Interface flyer memiliki 3 metode yang nantinya dapat diimplementasikan pada class yang membutuhkannya. Terdapat class airplane, bird, superman yang mengimplementasikan interface flyer dan masing-masing class nya meng over right metode interfacenya. Fungsi main terdapat sintax untuk menginisiasi class airplane, bird, superman dan menampilkan hasilnya.

3. Buat program berdasarkan UML berikut



Pemrograman :

```
public class prak6_no3 {
    public static void main(String[] args) {
        System.out.println("== Pesawat ==");
        Vehicle airplane_1 = new Vehicle();
        airplane_1.getLoad(20);
        airplane_1.setMaxLoad(max_load: 150);
        airplane_1.addBox( weight: 25);
        airplane_1.calcFuelEfficiency(200);
        airplane_1.calcTripDistance(30000);
        System.out.println("\n");

        Flyer airplane = new Airplane();
        System.out.println(airplane.takeOff());
        System.out.println(airplane.land());
        System.out.println(airplane.fly());

        System.out.println("\n== Burung ==");
        Bird bird = new Bird();
        System.out.println(bird.takeOff());
        System.out.println(bird.land());
        System.out.println(bird.fly());
        System.out.println(bird.buildNest());
        System.out.println(bird.layEggs());
    }
}
```

The screenshot shows the IntelliJ IDEA IDE interface with the project structure and code editor. The code implements the **Vehicle**, **Flyer**, and **Animal** interfaces. The **Vehicle** class has methods `getLoad()`, `setMaxLoad()`, `addBox()`, `calcFuelEfficiency()`, and `calcTripDistance()`. The **Airplane** class extends **Vehicle** and overrides the `takeOff()`, `land()`, and `fly()` methods. The **Bird** class extends **Flyer** and overrides the `takeOff()`, `land()`, `fly()`, `buildNest()`, and `layEggs()` methods.

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** pbo_prak6
- File:** prak6_no4.java
- Code:**

```
System.out.println(airplane.takeOff());
System.out.println(airplane.land());
System.out.println(airplane.fly());

System.out.println("\n==== Burung ===");
Bird bird = new Bird();
System.out.println(bird.takeOff());
System.out.println(bird.land());
System.out.println(bird.fly());
System.out.println(bird.buildNest());
System.out.println(bird.layEggs());
System.out.println(bird.eat( food: "Serangga"));

System.out.println("\n==== Superman ===");
Superman superman = new Superman();
System.out.println(superman.takeOff());
System.out.println(superman.land());
System.out.println(superman.fly());
System.out.println(superman.leapBuilding());
System.out.println(superman.stopBullet());
System.out.println(superman.tampil( name: "Mark Lee" , age: 24));
```
- Run:** prak6_no4
- Build:** Build completed successfully in 2 sec, 734 ms (12 minutes ago)
- System Bar:** Shows the date and time as 20/10/2022.

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** pbo_prak6
- File:** prak6_no4.java
- Code:**

```
interface Flyer{
    String takeOff();
    String land();
    String fly();

    1 usage 1 inheritor
    class Animal {
        1 usage
        String eat(String food) {
            return "Makanan Burung : " + food;
        }
    }

    1 usage 1 inheritor
    class Human {
        1 usage
        String tampil(String name, int age) {
            return "Nama : " + name + "\nUmur : " + age;
        }
    }
}
```
- Run:** prak6_no4
- Build:** Build completed successfully in 2 sec, 734 ms (12 minutes ago)
- System Bar:** Shows the date and time as 20/10/2022.

The screenshot shows the IntelliJ IDEA interface with the project 'pbo_prak6' open. The code editor displays two classes: `Human` and `Airplane`. The `Human` class has one usage of its `tampil` method. The `Airplane` class extends `Vehicle` and implements `Flyer`, overriding the `takeOff` and `land` methods. The code is written in Indonesian.

```
1 package pbo_prak6;
2
3 public class Human {
4     String tampil(String name, int age) {
5         return "Nama : " + name + "\nUmur : " + age;
6     }
7 }
8
9 public class Airplane extends Vehicle implements Flyer {
10
11     @Override
12     public String takeOff() {
13         return "+ Pesawat Lepas Landas Dari Bandara Soekarno Hatta Ke Bandara Incheon";
14     }
15
16     @Override
17     public String land() {
18         return "+ Pesawat Berhasil Mendarat Di Bandara Incheon";
19     }
20 }
```

The screenshot shows the IntelliJ IDEA interface with the project 'pbo_prak6' open. The code editor displays two classes: `Bird` and `Animal`. The `Bird` class extends `Animal` and implements `Flyer`, overriding the `fly`, `takeOff`, and `land` methods. The code is written in Indonesian.

```
1 package pbo_prak6;
2
3 public class Bird extends Animal implements Flyer{
4     @Override
5     public String fly() {
6         return "+ Burung Terbang Selama 7 Jam";
7     }
8
9     @Override
10    public String takeOff() {
11        return "+ Burung Kutting Terbang Dari Ketinggian 900 Meter Diatas Permukaan Laut";
12    }
13
14    @Override
15    public String land() {
16        return "+ Burung Kutting Mendarat Tepat Pada Atap Rumah Pak Andi";
17    }
18
19    @Override
20 }
```

The screenshot shows the IntelliJ IDEA interface with the project 'pbo_prak6' open. The code editor displays the file 'prak6_no3.java'. A code completion tooltip is visible over the class definition 'class Superman extends Human implements Flyer {'. The tooltip lists three usages of the class:

- 1 usage: `@Override` `public String fly() {`
- 1 usage: `return "+ Burung Kutilang Terbang Selama 45 Menit";`
- 1 usage: `}`
- 2 usages: `class Superman extends Human implements Flyer {`
- 3 usages: `@Override` `public String takeOff() {`
- return "+ Superman Terbang Dengan Cepat Menuju Tempat Terjadinya Pembunuhan";

The status bar at the bottom right shows the date and time: 14:48 CRLF UTF-8 4 spaces 19.27 20/10/2022.

The screenshot shows the IntelliJ IDEA interface with the project 'pbo_prak6' open. The code editor displays the file 'prak6_no3.java'. A code completion tooltip is visible over the class definition 'class Superman extends Human implements Flyer {'. The tooltip lists five usages of the class:

- 3 usages: `@Override` `public String takeOff() {`
- return "+ Superman Terbang Dengan Cepat Menuju Tempat Terjadinya Pembunuhan";
- 3 usages: `}`
- 3 usages: `@Override` `public String land() {`
- return "+ Superman Mendarat Di Tempat Terjadinya Pembunuhan";
- 3 usages: `}`
- 3 usages: `@Override` `public String fly() {`
- return "+ Superman Terbang Dengan Kecepatan 80 Km/jam";
- 3 usages: `}`
- 1 usage: `public String leapBuilding() {`
- return "+ Superman Terbang Melewati Banyak Gedung Tinggi";
- 1 usage: `}`

The status bar at the bottom right shows the date and time: 14:48 CRLF UTF-8 4 spaces 19.27 20/10/2022.

The screenshot shows the IntelliJ IDEA interface. The code editor displays a Java file named prak6_no3.java. The code defines a class Superman with four methods: fly(), leapBuilding(), stopBullet(), and main(). The main() method calls the other three methods. The code uses annotations like @Override and includes comments describing the actions of Superman. The run tab at the bottom shows the output of the program.

```
pbo_prak6 - prak6_no3.java
prak6_no3.java
109     return "+ Superman Mendarat Di Tempat Terjadinya Pembunuhan";
110 }
111
112     3 usages
113     @Override
114     public String fly() {
115         return "+ Superman Terbang Dengan Kecepatan 80 Km/jam";
116     }
117
118     1 usage
119     public String leapBuilding() {
120         return "+ Superman Terbang Melewati Banyak Gedung Tinggi";
121     }
122
123     1 usage
124     public String stopBullet() {
125         return "+ Superman Tidak Bisa Ditembak Peluru";
126     }
127 }

Run: prak6_no4
```

Hasil :

The screenshot shows the IntelliJ IDEA interface with the run tab selected. It displays the output of the program prak6_no3. The output shows the results for Pesawat, Burung, and Superman, each with their respective actions listed. The run tab also shows the command used to run the program and the build status.

```
prak6_no3.x
" C:\Program Files\Eclipse Adoptium\jdk-17.0.4.101-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8
== Pesawat ==
Load : 20.0
Max Load : 150.0
Add Box : 25.0
Fuel Efficiency : 200.0
Trip Distance : 30000.0

+ Pesawat Lepas Landas Dari Bandara Soekarno Hatta Ke Bandara Incheon
+ Pesawat Berhasil Mendarat Di Bandara Incheon
+ Pesawat Terbang Selama 7 Jam

== Burung ==
+ Burung Kutilang Terbang Dari Ketinggian 900 Meter Diatas Permukaan Laut
+ Burung Kutilang Mendarat Tepat Pada Atap Rumah Pak Andi
+ Burung Kutilang Terbang Selama 45 Menit
+ Burung Kutilang Membangun Sarangnya Di Batang Pohon Yang Tertutup
+ Burung Kutilang Menggerami Telurnya
Makanan Burung : Serangga

== Superman ==
+ Superman Terbang Dengan Cepat Menuju Tempat Terjadinya Pembunuhan
+ Superman Mendarat Di Tempat Terjadinya Pembunuhan
```

The screenshot shows an IDE interface with the following details:

- Project:** prak6
- Run:** prak6_no3
- Output Window Content:**

```
+ Pesawat Lepas Landas Dari Bandara Soekarno Hatta Ke Bandara Incheon
+ Pesawat Berhasil Mendarat Di Bandara Incheon
+ Pesawat Terbang Selama 7 Jam

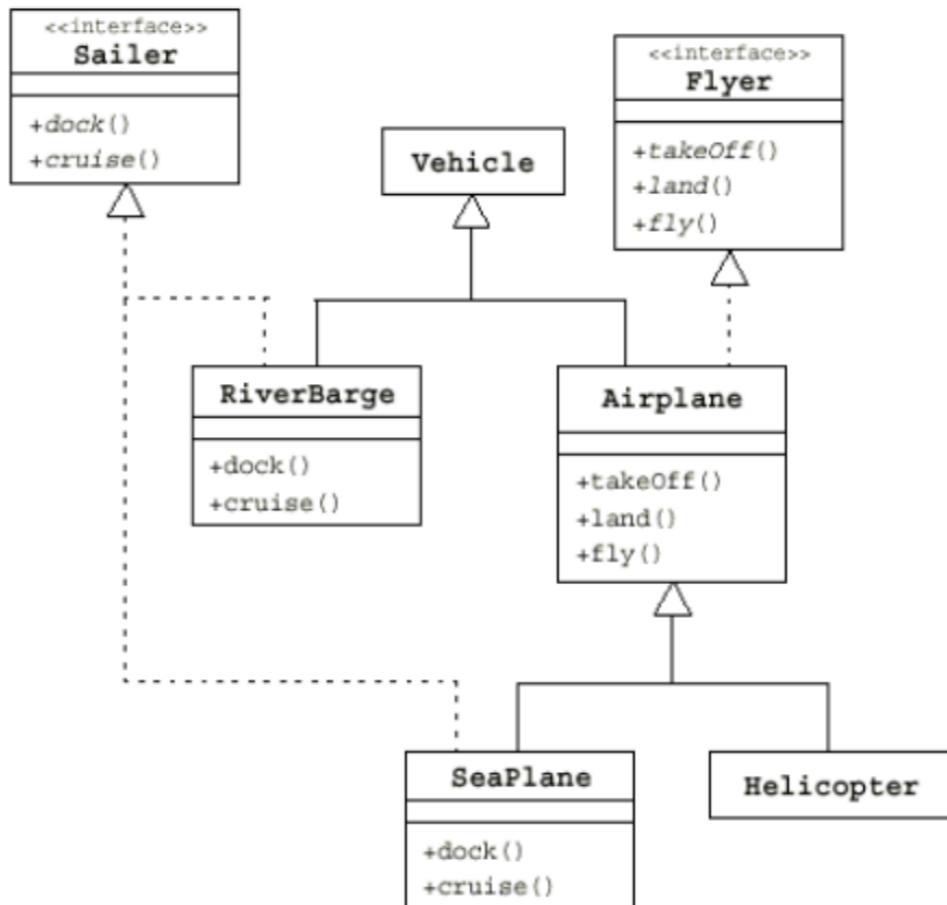
== Burung ==
+ Burung Kutilang Terbang Dari Ketinggian 900 Meter Diatas Permukaan Laut
+ Burung Kutilang Mendarat Tepat Pada Atap Rumah Pak Andi
+ Burung Kutilang Terbang Selama 45 Menit
+ Burung Kutilang Membangun Sarangnya Di Batang Pohon Yang Tertutup
+ Burung Kutilang Menggerami Telurnya
Makanan Burung : Serangga

== Superman ==
+ Superman Terbang Dengan Cepat Menuju Tempat Terjadinya Pembunuhan
+ Superman Mendarat Di Tempat Terjadinya Pembunuhan
+ Superman Terbang Dengan Kecepatan 80 Km/jam
+ Superman Terbang Melalui Banyak Gedung Tinggi
+ Superman Tidak Bisa Ditembak Peluru
Nama : Mark Lee
Umur : 24

Process finished with exit code 0
```
- Bottom Status Bar:**
 - Build completed successfully in 3 sec, 354 ms (a minute ago)
 - 114:64 CRLF UTF-8 4 spaces
 - 19.01 20/10/2022

Analisa : Sama seperti program-program sebelumnya kode diatas memiliki interface flyer yang memiliki 3 metode yang sama. Kode prgram diatas terdapat 2 class tambahan yaitu class animal dan human. Berbeda dengan kode sebelumnya class airplane, bird, dan superman mewarisi dari class induknya. Class airplane mewarisi class vehicle yang terdapat pada soal no 1. Class bird mewarisi class animal dan class superman mewarisi class human. Inisialisasi dan hasil ditampilkan pada fungsi main.

4. Buat program berdasarkan UML berikut



The screenshot shows an IDE interface with the project structure and code editor. The code editor displays Java code for the **RiverBarge** and **SeaPlane** classes.

```

public class prak6_no4 {
    public static void main(String[] args) {
        RiverBarge riverBarge_1 = new RiverBarge();

        System.out.println("== River Barge ==");
        riverBarge_1.getLoad(25);
        riverBarge_1.getMaxLoad(max_load: 250);
        riverBarge_1.addBox(weight: 40.5);
        System.out.println(riverBarge_1.dock());
        System.out.println(riverBarge_1.cruise());
        System.out.println("\n");

        System.out.println("== Sea Plane ==");
        SeaPlane seaPlane = new SeaPlane();
        System.out.println(seaPlane.dock());
        System.out.println(seaPlane.cruise());
        System.out.println("\n");

        System.out.println("== Airplane ==");
        Vehicle airplane_1 = new Vehicle();
        airplane_1.getLoad(20);
        airplane_1.getMaxLoad(max_load: 150);
        airplane_1.addBox(weight: 25);
        airplane_1.calcFuelEfficiency(200);
    }
}
  
```

The code demonstrates the creation of objects for **RiverBarge**, **SeaPlane**, and **Vehicle** (representing **Airplane**) and calls their respective methods **dock()**, **cruise()**, and **getLoad()**.

The screenshot shows the IntelliJ IDEA interface with the project 'pbo_prak6' open. The code editor displays the file 'prak6_no4.java'. The code implements the Flyer interface with methods takeOff(), land(), and fly(). It also defines a Vehicle class with a double load attribute and a getLoad(double load) method. The code uses System.out.println statements to output the airplane's name and trip details.

```
System.out.println("== Airplane ==");
Vehicle airplane_1 = new Vehicle();
airplane_1.getLoad(20);
airplane_1.setMaxLoad(max_load: 150);
airplane_1.addBox( weight: 25);
airplane_1.calcFuelEfficiency(200);
airplane_1.calcTripDistance(30000);
System.out.println("\n");

Flyer airplane = new Airplane();
System.out.println(airplane.takeOff());
System.out.println(airplane.land());
System.out.println(airplane.fly());

}

2 usages 2 implementations
interface Flyer{
    1 usage 1 implementation
    String takeOff();
    1 usage 1 implementation
    String land();
    1 usage 1 implementation
    String fly();
}
```

This screenshot shows the same IntelliJ IDEA interface, but the code editor now displays the definition of the Flyer interface and its implementation. The Vehicle class is also partially visible at the bottom of the code editor.

```
2 usages 2 implementations
interface Flyer{
    1 usage 1 implementation
    String takeOff();
    1 usage 1 implementation
    String land();
    1 usage 1 implementation
    String fly();
}

2 usages 2 implementations
interface Sailer {
    2 usages 2 implementations
    String dock();
    2 usages 2 implementations
    String cruise();
}

4 usages 3 inheritors
class Vehicle{
    double load = 0;
    double max_load = 0;

    2 usages
    void getLoad(double load) {
```

The screenshot shows the IntelliJ IDEA interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** prak6, src, prak6_no4.java, RiverBarge, Current File.
- Project Tree:** prak6_no1.java, prak6_no2.java, prak6_no3.java, prak6_no4.java, prak6_no4.iml, External Libraries, Scratches and Consoles.
- Code Editor:** The active file is prak6_no4.java, displaying the following code:

```
4 usages 3 inheritors
45 class Vehicle{
46     double load = 0;
47     double max_load = 0;
48
49     2 usages
50     void getLoad(double load) {
51         System.out.println("Load : " + load);
52     }
53     2 usages
54     void getMaxLoad(double max_load) {
55         System.out.println("Max Load : " + max_load);
56     };
57     2 usages
58     void addBox(double weight) {
59         System.out.println("Add Box : " + weight);
60     };
61     1 usage
62     public void calcFuelEfficiency(double fuel) {
63         System.out.println("Fuel Efficiency : " + fuel);
64     }
65 }
```

- Run Bar:** Run, prak6_no4.
- Bottom Status Bar:** Build completed successfully in 2 sec, 734 ms (moments ago), 107:48, CRLF, UTF-8, 4 spaces, 19.15, 20/10/2022.

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Tree:** The project is named "prak6_no4.java". It contains a "src" directory with files: prak6_no1.java, prak6_no2.java, prak6_no3.java, and prak6_no4.java.
- Code Editor:** The current file is "prak6_no4.java". The code defines a class "Airplane" that extends "Vehicle" and implements "Flyer".

```
1 usage
void calcTripDistance(double trip) {
    System.out.println("Trip Distance : " + trip);
}

2 usages 1 inheritor
class Airplane extends Vehicle implements Flyer {

    1 usage
    @Override
    public String takeOff() {
        return "+ Pesawat Lepas Landas Dari Bandara Soekarno Hatta Ke Bandara Incheon";
    }

    1 usage
    @Override
    public String land() {
        return "+ Pesawat Berhasil Mendarat Di Bandara Incheon";
    }

    1 usage
    @Override
    public String fly() {
```
- Toolbars and Status Bar:** The top bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and a "Current File" dropdown. The bottom bar shows tabs for Version Control, Run, TODO, Problems, Terminal, Services, and Build. The status bar at the bottom right shows the time as 10:48, CRLF, UTF-8, 4 spaces, and the date as 20/10/2023.

The screenshot shows the IntelliJ IDEA interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** pbo_prak6 - prak6_no4.java, RiverBarge, cruise.
- Toolbars:** Standard toolbar with icons for Open, Save, Run, Stop, etc.
- Left Sidebar:** Project (pbo_prak6), External Libraries, Scratches and Consoles.
- Code Editor:** The editor displays the file prak6_no4.java with the following code:

```
81     @Override
82     public String fly() {
83         return "+ Pesawat Terbang Selama 7 Jam";
84     }
85
86
87     class SeaPlane extends Airplane implements Sailer {
88         @Override
89         public String dock() {
90             return "Jumlah Dermaga Seaplane, 5";
91         }
92
93         @Override
94         public String cruise() {
95             return "Jumlah Pelayaran Seaplane, 3";
96         }
97     }
98
99     class RiverBarge extends Vehicle implements Sailer {
```

The code editor highlights several parts of the code in orange, indicating they are part of the current file's scope or have been used elsewhere.

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Bar:** Shows the project name "pbo_prak6" and the file "prak6_no4.java" is open.
- Toolbars:** Standard IntelliJ toolbars for File, Edit, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Code Editor:** The code for the `RiverBarge` class is displayed. The class extends `Vehicle` and implements `Sailor`. It overrides the `cruise()` and `dock()` methods.

```
public String cruise() {
    return "Jumlah Pelayaran Seaplane, 3";
}

class RiverBarge extends Vehicle implements Sailor {
    public String dock() {
        return "Jumlah Dermaga River Barge, 8";
    }

    public String cruise() {
        return "Jumlah Pelayaran River Barge, 5";
    }
}
```

- Run Tab:** Shows the run configuration "prak6_no4".
- Bottom Status Bar:** Displays build status, time (10:48), encoding (CRLF), file format (UTF-8), and line count (4 spaces).

Hasil :

```
"C:\Program Files\Eclipse Adoptium\jdk-17.0.4.101-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8 prak6_no4
prak6_no4
Run: prak6_no4 x
Load : 25.0
Max Load : 250.0
Add Box : 40.5
Jumlah Dermaga River Barge, 8
Jumlah Pelayaran River Barge, 5

== Sea Plane ==
Jumlah Dermaga Seaplane, 5
Jumlah Pelayaran Seaplane, 3

== Airplane ==
Load : 20.0
Max Load : 150.0
Add Box : 25.0
Fuel Efficiency : 200.0
Trip Distance : 30000.0

+ Pesawat Lepas Landas Dari Bandara Soekarno Hatta Ke Bandara Incheon
+ Pesawat Berhasil Mendarat Di Bandara Incheon

Build completed successfully in 2 sec, 734 ms (3 minutes ago)
107:48 CRLF UTF-8 4 spaces
19.18 IND 20/10/2022
```

```
Jumlah Dermaga River Barge, 8
Jumlah Pelayaran River Barge, 5

== Sea Plane ==
Jumlah Dermaga Seaplane, 5
Jumlah Pelayaran Seaplane, 3

== Airplane ==
Load : 20.0
Max Load : 150.0
Add Box : 25.0
Fuel Efficiency : 200.0
Trip Distance : 30000.0

+ Pesawat Lepas Landas Dari Bandara Soekarno Hatta Ke Bandara Incheon
+ Pesawat Berhasil Mendarat Di Bandara Incheon
+ Pesawat Terbang Selama 7 Jam

Process finished with exit code 0

Build completed successfully in 2 sec, 734 ms (4 minutes ago)
107:48 CRLF UTF-8 4 spaces
19.18 IND 20/10/2022
```

Analisa : pada program diatas memiliki 2 interface yaitu flyer dan sailer. Class vehicle memiliki 2 variabel dan 5 metode yang nantinya akan digunakan pada class turunannya. Class airplane, sea plane, riverbarge sama-sama mewarisi class induknya dan mengimplementasi class interface yang telah dibuat. Inisialisasi class dan pengiriman data pada metode terdapat di fungsi main.