

**ULANGAN AKHIR SEMESTER
PEMROGRAMAN BERORIENTASI OBJEK**



Disusun Oleh :

Nama : Yoga Pramudita

NIM : L200200182

Kelas : C

Matkul : PBO

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2021/2022**

1. a. Buatlah suatu class yang bernama Animal yang terdapat non-void parameterized method overloading.
b. Kemudian, tambahkan satu class lain bernama Cat yang meng-override terhadap seluruh method pada class 1a.

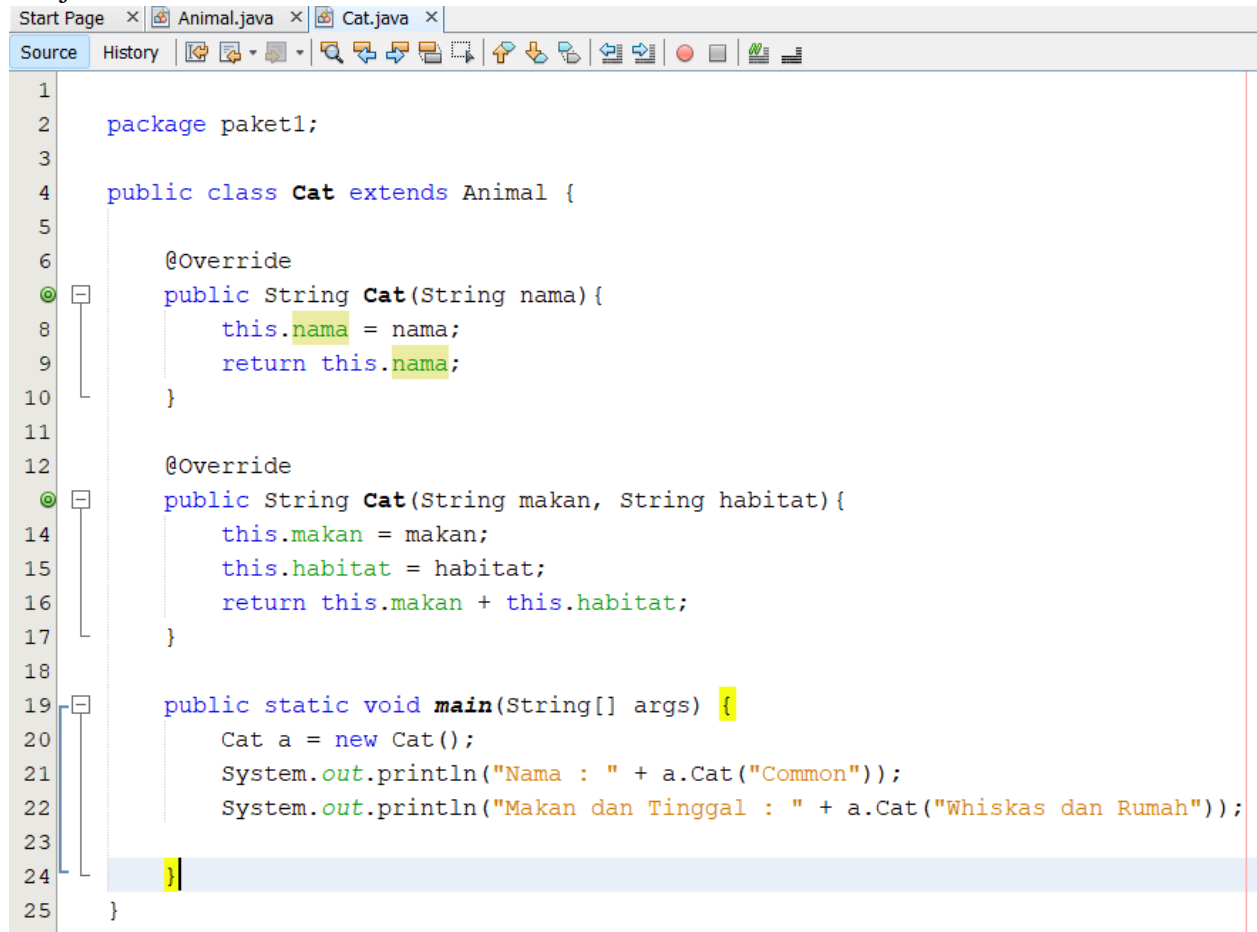
Jawab :

Screenshot Program :

a. Animal.java

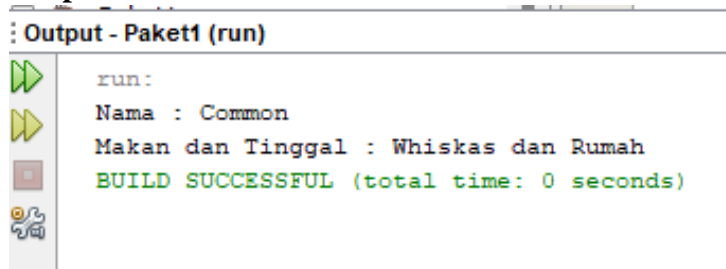
```
1
2     package paket1;
3
4     public class Animal {
5         public String nama;
6         public String makan;
7         public String habitat;
8
9         public String Kucing(String nama) {
10             this.makan = makan;
11             return this.nama;
12         }
13
14         public String Kucing(String makan, String habitat){
15             this.makan = makan;
16             this.habitat = habitat;
17             return this.makan + this.habitat;
18         }
19     }
```

b. Cat.java



```
1
2 package paket1;
3
4 public class Cat extends Animal {
5
6     @Override
7     public String Cat(String nama){
8         this.nama = nama;
9         return this.nama;
10    }
11
12    @Override
13    public String Cat(String makan, String habitat){
14        this.makan = makan;
15        this.habitat = habitat;
16        return this.makan + this.habitat;
17    }
18
19    public static void main(String[] args) {
20        Cat a = new Cat();
21        System.out.println("Nama : " + a.Cat("Common"));
22        System.out.println("Makan dan Tinggal : " + a.Cat("Whiskas dan Rumah"));
23    }
24 }
25
```

Output :



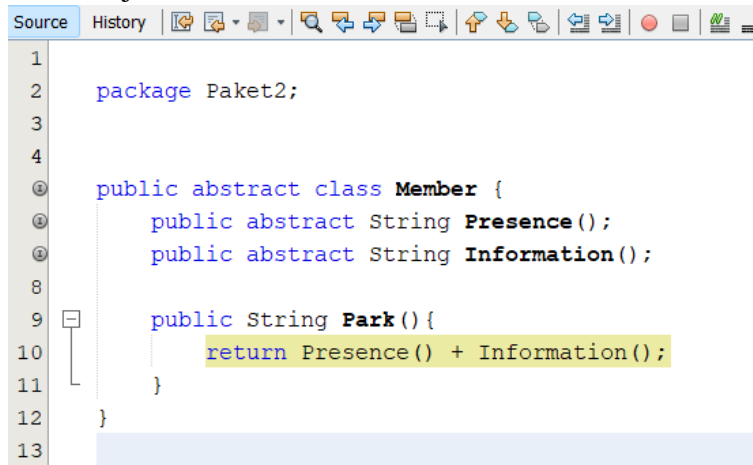
```
run:
Nama : Common
Makan dan Tinggal : Whiskas dan Rumah
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. a. Buatlah sebuah abstract class bernama Member yang terdapat:
- Dua abstract methods: presence dan information
 - Satu concrete method: park
- b. Kemudian, buatlah subclass dari abstract class tersebut. Apa yang anda lakukan supaya program anda dapat berjalan dengan baik?

Jawab :

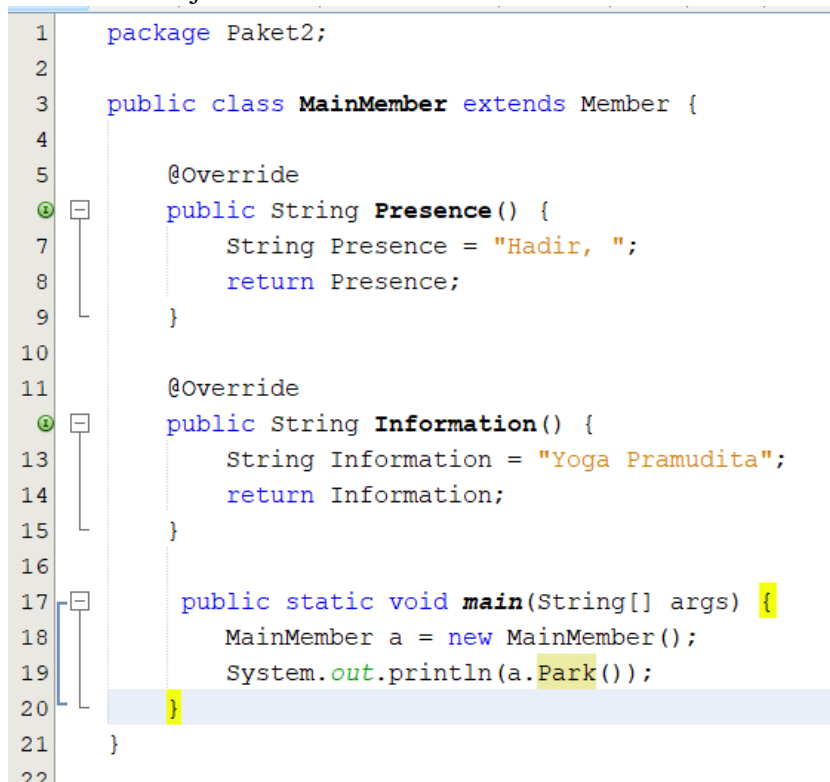
Screenshot Program :

a. Member.java



```
Source History
1
2 package Paket2;
3
4
5 public abstract class Member {
6     public abstract String Presence();
7     public abstract String Information();
8
9     public String Park() {
10         return Presence() + Information();
11     }
12 }
13
```

b. MainMember.java



```
1 package Paket2;
2
3 public class MainMember extends Member {
4
5     @Override
6     public String Presence() {
7         String Presence = "Hadir, ";
8         return Presence;
9     }
10
11     @Override
12     public String Information() {
13         String Information = "Yoga Pramudita";
14         return Information;
15     }
16
17     public static void main(String[] args) {
18         MainMember a = new MainMember();
19         System.out.println(a.Park());
20     }
21 }
22
```

Output :

```
Output - Paket2 (run)

run:
Hadir, Yoga Pramudita
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. a. Buatlah sebuah interface yang bernama Action yang terdapat satu method bernama Move.
- b. Buatlah sebuah interface lain bernama Walk yang merupakan sub interface dari interface 3a. Interface 3b memiliki satu variable array yang menampung daftar arah yang tersedia: forward, backward, left, right – beserta satu method bernama SpeedUp.
- c. Buatlah suatu class Player beserta isinya, yang mengimplementasikan subinterface.

Jawab :

Screenshot Program :

a. Action.java

```
1
2 package paket3;
3 interface Action{
4     public void Move ();
5 }
6
7
```

b. Walk.java

```
Source History [Icons]
1
2 package paket3;
3
4
5 interface Walk extends Action{
6     String [] action = {"forward", "backward", "left", "right"};
7     public void SpeedUp ();
8
9 }
10
```

c.Player.java

```
1
2 package paket3;
3
4
5 public class Player implements Walk {
6
7     @Override
8     public void SpeedUp() {
9         System.out.println("Player Can Move");
10    }
11
12    @Override
13    public void Move() {
14        System.out.println("Move" + action [0] + "\n" +
15                           "Move" + action [1] + "\n" +
16                           "Move" + action [2] + "\n" +
17                           "Move" + action [3]);
18    }
19
20    public static void main(String[] args) {
21        Player a = new Player();
22        a.SpeedUp();
23        a.Move();
24    }
25 }
26
```

Output :

```
Output - Paket3 (run)
run:|
Player Can Move
Moveforward
Movebackward
Moveleft
Moveright
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. a. Buatlah suatu outer class yang memiliki:
- Sebuah instance variable untuk code; sebuah static variable untuk count.
 - Sebuah constructor untuk menghitung jumlah object yang telah dibuat dari class yang bersangkutan.
 - Satu inner class dan satu static nested class dimana setiap class 15 2 memiliki satu variable dan satu method.

- Di dalam method pada inner class panggillah instance variable yang ada di outer class.
- Di dalam method pada static nested class, panggillah static variable yang ada di outer class.

b. Kemudian, buatlah suatu demo class yang berisikan main method. Di dalam main method terdapat perintah untuk membuat suatu object:

- outerObj dari Outer Class
- innerObj dari Inner Class
- nestedObj dari Static Nested Class

Jawab :

a. Outer.java

```

1
2  package paket4;
3
4  public class Outer {
5      public String code = "NIM : L200200182";
6      public static int count;
7
8      Outer() {
9          count += 1;
10         System.out.println("Outer: " + count);
11     }
12
13     class Inner{
14         String inner = "this is Inner Class";
15         String inner() {
16             return code;
17         }
18     }
19     static class Nested{
20         String nested = "this is Nested Class";
21         int nested() {
22             return count;
23         }
24     }
25 }
26
27

```

b. DemoOuter.java

```
1
2  package paket4;
3
4  public class Outer {
5      public String code = "NIM : L200200182";
6      public static int count;
7
8      Outer() {
9          count +=1;
10         System.out.println("Outer: " + "count");
11     }
12
13     class Inner{
14         String inner = "this is Inner Class";
15         String inner() {
16             return code;
17         }
18     }
19     static class Nested{
20         String nested = "this is Nested Class";
21         int nested() {
22             return count;
23         }
24     }
25
26 }
27
```

Output :

```
Output - Paket4 (run)
run:
Outer: 1
NIM : L200200182
this is Inner Class
this is Nested Class
BUILD SUCCESSFUL (total time: 0 seconds)
```


5. a. Tuliskan code untuk mengambil nilai dari TextBox?
b. Tuliskan code untuk menampilkan nilai ke Label?

Jawab :

a.

```
JTextArea a = new JTextArea("L200200182");  
String nim = a.getText();
```

b.

```
JLabel lb = new JLabel(nim);
```