

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

BAB : INTERFACE
NAMA : Rochmanu Purnomohadi Erfitra
NIM : 225150200111018
ASISTEN : Tengku Muhammad Rafi Rahardiansyah
Muhammad Bin Djafar Almasyyhur
TGL PRAKTIKUM : 10 Mei 2023

A. Interface

```
1 package praktikumpl;
2
3 public interface Colorable {
4     public void howToColor();
5 }
6 public interface Comparable
7
8 {
9     public void compareTo(Object obj);
10 }
11 public class Rectangle implements Colorable, Comparable{ //
12     lass rectanggle
13     private String warna;
14     private int kategori;
15
16     public Rectangle() {
17     }
18
19     public Rectangle(String warna) {
20         this.warna = warna;
21     }
22
23     public void howToColor() {
24         if(this.warna == null){
25             System.out.println("tidak ada warna, warna bangun kotak
26 masih polos");
27         }
28         else{
29             System.out.println("bangun kotak sudah diwarnai dengan
30 warna "+this.warna);
31         }
32     }
33
34     public void compareTo(Object obj) {
35         this.kategori = (int) obj;
36         if(this.kategori == 0){
37             System.out.println("ukuran cat yang cocok untuk bangun
38 kotak dengan ukuran kategori " +this.kategori+" yaitu 2.5L"
39 ); }
40         else{
41             System.out.println("ukuran cat yang cocok untuk bangun
42 kotak dengan ukuran kategori " +this.kategori+" yaitu 6.5L" ); }
43     }
44 }
```

```

45 public static void main(String[] args) {
46
47     Rectangle kotak1 = new Rectangle("merah");
48     Rectangle kotak2= new Rectangle();
49     Rectangle kotak3 = new Rectangle();
50     kotak1.howToColor();
51     kotak2.howToColor();
52     kotak3.compareTo(4);
53
54 }
55

```

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan serta jelaskan!

Syntax

```

interface Colorable {

    public void howToColor();

}

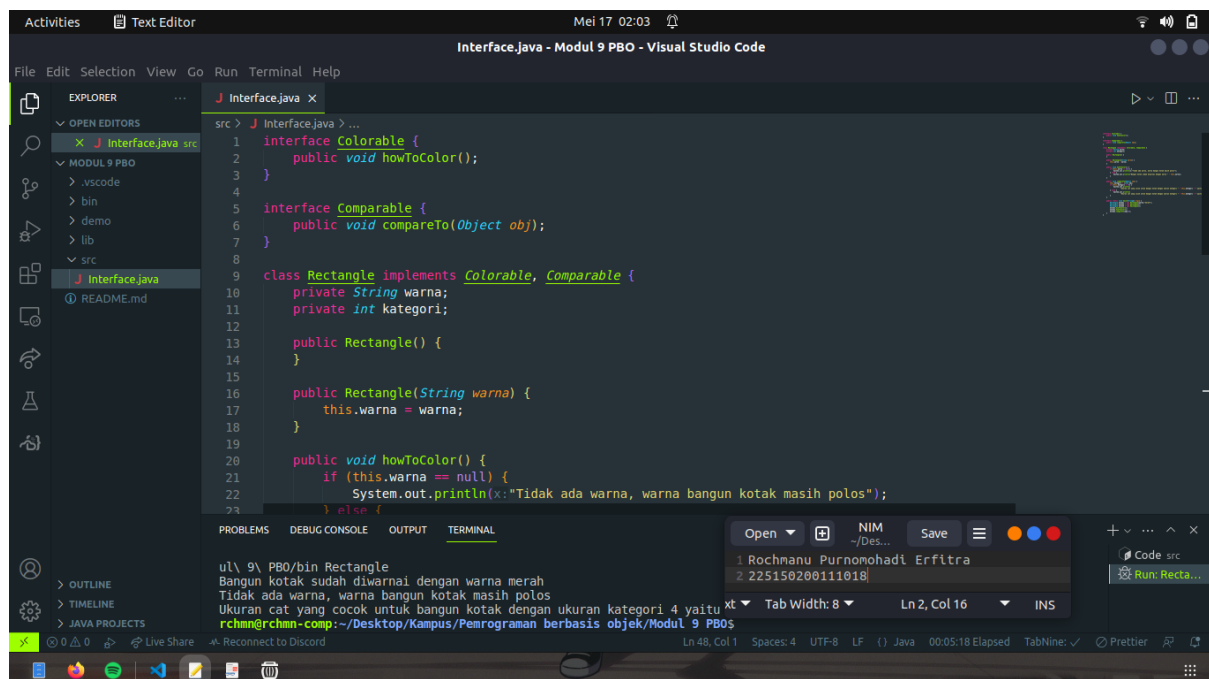
interface Comparable {

    public void compareTo(Object obj);

}

```

Screenshot



Penjelasan

Main method berada diluar main class maka perlu sedikit penyesuaian, dan karena method interface berada bersama dalam satu file dengan main class maka keduanya tidak boleh memiliki access modifier public yang akan menyebabkan konflik, jika memang perlu access modifier public maka harus dinyatakan dalam file yang berbeda.

Pertanyaan

2. Apakah class yang berbentuk Interface bisa diinstansiasi menjadi sebuah objek? Jelaskan alasannya!

Penjelasan

Tidak, karena pada interface hanya berisi kerangka yang nantinya akan diimplementasi pada class yang menggunakan interface tersebut jadi

Pertanyaan

3. Apakah suatu class dapat mengimplementasi class interface yang jumlahnya lebih dari satu? Jelaskan alasannya!

Syntax

```
class Rectangle implements Colorable, Comparable {  
    private String warna;  
    private int kategori;  
  
    public Rectangle() {  
    }  
  
    public Rectangle(String warna) {  
        this.warna = warna;  
        System.out.println("nilai pengurangan adalah : " +  
nilai);  
    }  
}
```

Penjelasan

Bisa, namun perlu diperhatikan semua implementasi dalam interface harus ada dalam class tersebut. Perlu dicatat, yang diwajibkan hanya satu pewarisan hanyalah pada konsep inheritance dimana child class hanya boleh mewarisi satu parent class saja

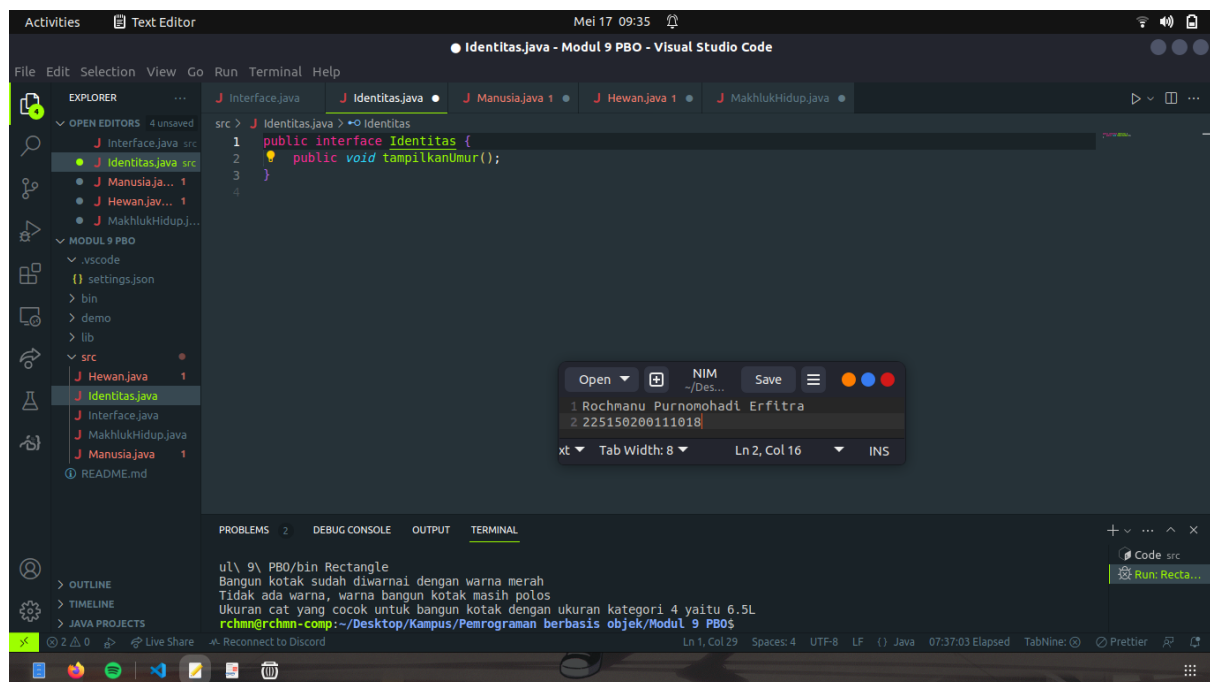
Pertanyaan

4. Pada interface Identitas.java hapus method tampilkan nama, amati apa yang terjadi dan mengapa demikian?

Syntax

```
public interface Identitas {  
    public void tampilkanUmur();  
}
```

Screenshot



Penjelasan

Jika method tampilkanNama() dihapus dari interface Identitas.java, maka class-class yang mengimplementasikan interface tersebut akan menghasilkan kesalahan kompilasi (compilation error). Hal ini terjadi karena class-class tersebut harus memberikan implementasi untuk semua metode yang didefinisikan dalam interface yang telah dihapus tadi.

Pertanyaan

5. Jika pada class hewan kita hanya ingin mengimplements interface MakhlukHidup saja apa yang terjadi? Jelaskan

Syntax

```
public class Hewan implements MakhlukHidup {

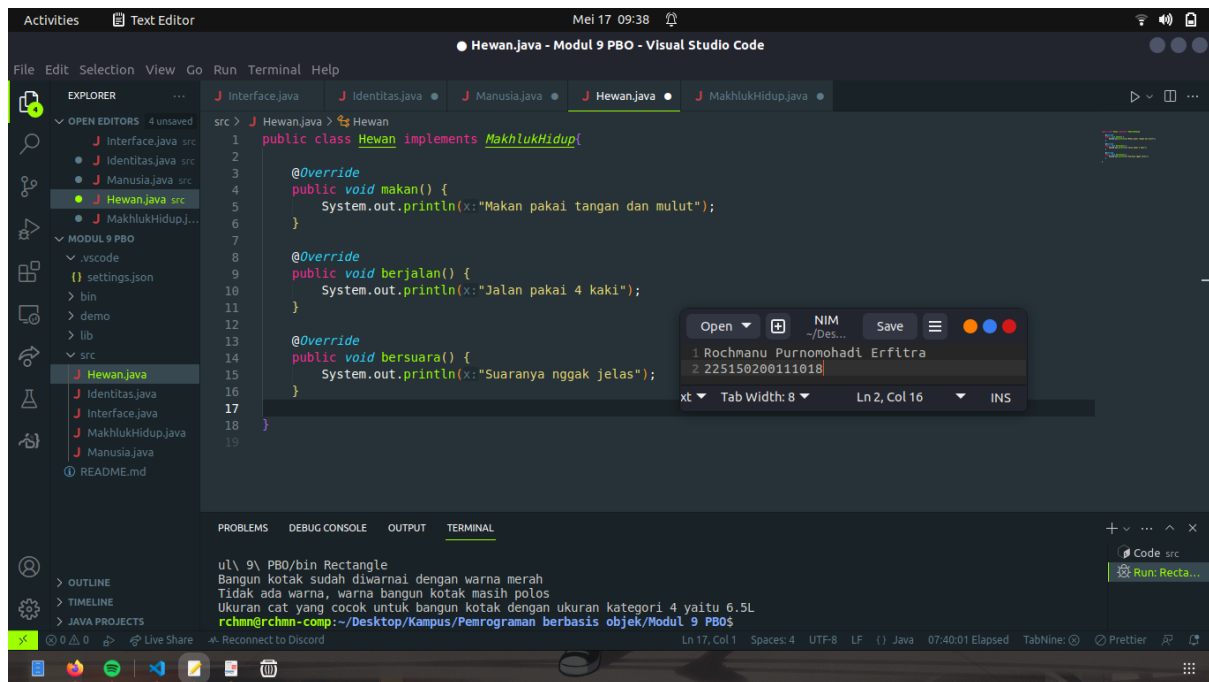
    @Override
    public void makan() {
        System.out.println("Makan pakai tangan dan mulut");
    }

    @Override
    public void berjalan() {
        System.out.println("Jalan pakai 4 kaki");
    }

    @Override
    public void bersuara() {
        System.out.println("Suaranya nggak jelas");
    }

}
```

Screenshot



Penjelasan

Dengan menghapus implementasi Identitas pada class Hewan, kita tidak perlu menyediakan implementasi untuk method tampilkanNama() dan tampilkanUmur().

Pertanyaan

6. Buatlah konstruktor pada manusia dengan parameter umur dan nama kemudian panggil pada Class Main dengan menginstan objek bernama nama anda!

Syntax

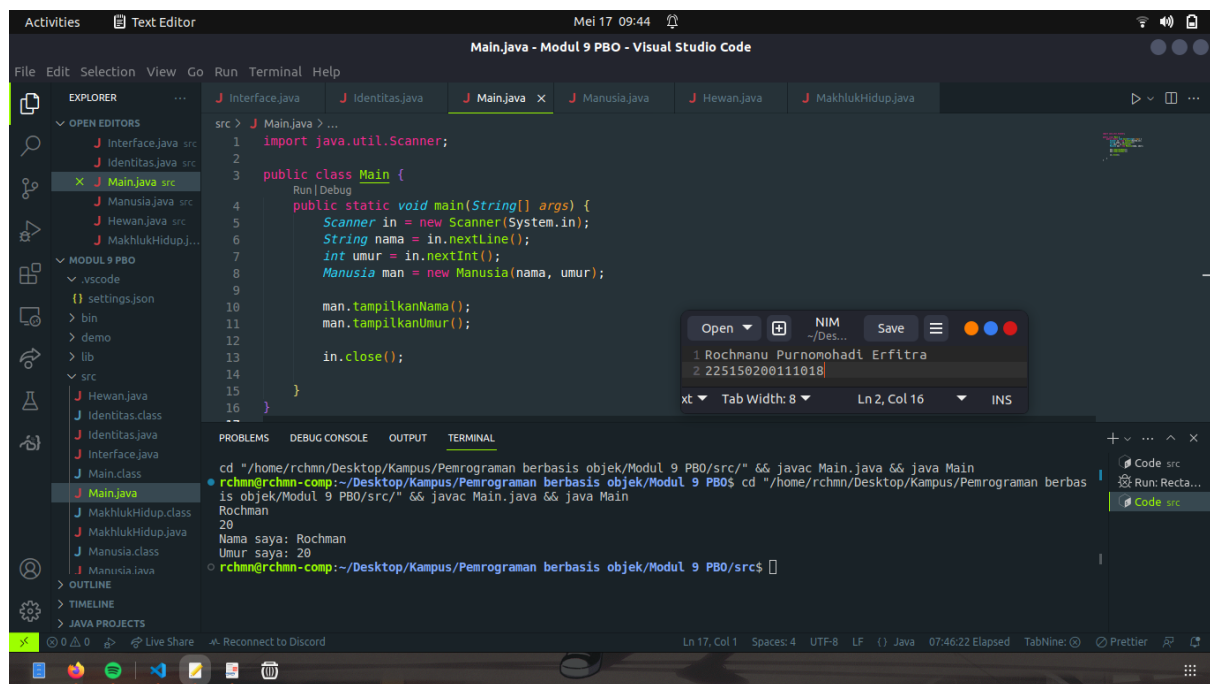
	<pre>import java.util.Scanner; public class Main { public static void main(String[] args) { Scanner in = new Scanner(System.in); String nama = in.nextLine(); int umur = in.nextInt(); Manusia man = new Manusia(nama, umur); man.tampilkanNama(); man.tampilkanUmur(); } }</pre>
--	---

```
        in.close();

    }

}
```

Screenshot



Penjelasan

Ada pada code

Pertanyaan

- Ubah source code diatas menjadi proses meminta inputan dari user dan buat menjadi interaktif!

Syntax

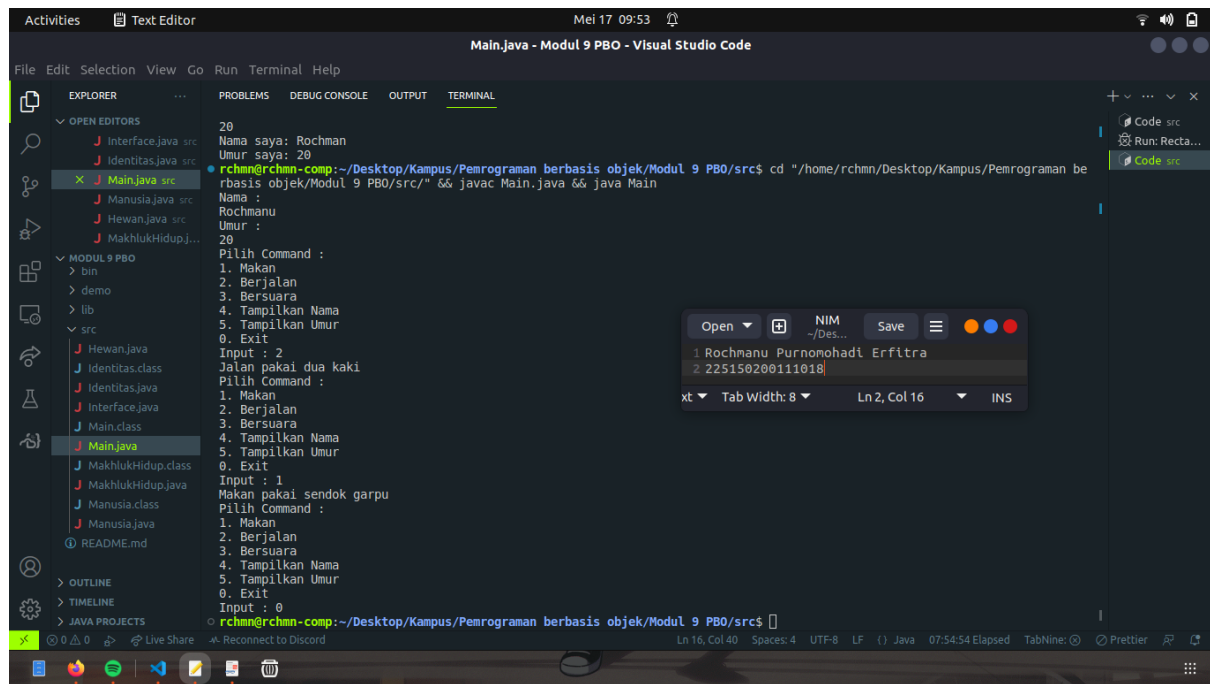
```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int ext = 1;
        System.out.println("Nama : ");
        String nama = in.nextLine();
        System.out.println("Umur : ");
        int umur = in.nextInt();
        in.nextLine();

        Manusia man = new Manusia(nama, umur);
        while (ext != 0) {
            System.out.print("Pilih Command : " +
                "\n" + "1. Makan" +
                "\n" + "2. Berjalan" +
                "\n" + "3. Bersuara" +
                "\n" + "4. Tampilkan Nama" +
                "\n" + "5. Tampilkan Umur" +
                "\n" + "0. Exit" +
                "\n" + "Input : ");
            ext = in.nextInt();
            switch (ext){
                case 1:
                    man.makan();
                    break;
```


	<pre> case 2: man.berjalan(); break; case 3: man.bersuara(); break; case 4: man.tampilkanNama(); break; case 5: man.tampilkanUmur(); break; } } in.close(); }</pre>
	}

Screenshot



Penjelasan

Ada pada code

Pertanyaan

8. Buat objek selain objek diatas dengan menggunakan method yang berbeda dengan yang diatas! (min.1 contoh)

Syntax

	<pre> public class Tumbuhan implements MakhlukHidup, Identitas { private String nama; private int umur; public Tumbuhan(String nama) { this.nama = nama; } @Override public void makan() { </pre>
--	--

```
        System.out.println("Tumbuhan melakukan
fotosintesis");
    }

    @Override
    public void berjalan() {
        System.out.println("Tumbuhan tidak dapat
berjalan");
    }

    @Override
    public void bersuara() {
        System.out.println("Tumbuhan tidak bersuara");
    }

    @Override
    public void tampilkanNama() {
        System.out.println("Nama tumbuhan: " + this.nama);
    }

    @Override
    public void tampilkanUmur() {
        System.out.println("Umur tumbuhan: " + this.umur);
    }
}
```

Screenshot

```
File Edit Selection View Go Run Terminal Help

Main.java - Modul 9 PBO - Visual Studio Code

EXPLORER
  OPEN EDITORS
    Interface.java src
    Identitas.java src
    Main.java src
    Manusia.java src
    Hewan.java src
    Tumbuhan.java...
    MakhlukHidup.j...
  MODUL 9 PBO
    demo
    lib
    src
      Hewan.java
      Identitas.class
      Identitas.java
      Interface.java
      Main.class
      Main.java
      MakhlukHidup.class
      MakhlukHidup.java
      Manusia.class
      Manusia.java
      Tumbuhan.class
      Tumbuhan.java
    OUTLINE
    TIMELINE
    JAVA PROJECTS

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

rchnn@rchnn-comp:~/Desktop/Kampus/Pemrograman berbasis objek/Modul 9 PBO/src$ cd "/home/rchnn/Desktop/Kampus/Pemrograman be
rbasis objek/Modul 9 PBO/src/" && javac Main.java && java Main
Nama Tumbuhan :
Anggrek
Pilih Command :
1. Makan
2. Berjalan
3. Bersuara
4. Tampilkan Nama
0. Exit
Input : 1
Tumbuhan melakukan fotosintesis
Pilih Command :
1. Makan
2. Berjalan
3. Bersuara
4. Tampilkan Nama
0. Exit
Input : 2
Tumbuhan tidak dapat berjalan
Pilih Command :
1. Makan
2. Berjalan
3. Bersuara
4. Tampilkan Nama
0. Exit
Input : 0
rchnn@rchnn-comp:~/Desktop/Kampus/Pemrograman berbasis objek/Modul 9 PBO/src$

Open *NIM Save
~/Des...
1 Rochmanu Purnomohadi Erfitra
2 225150200111018
xt Tab Width: 8 Ln2, Col 17 INS
```

Penjelasan

Ada pada code