

BAB 3

OVERLOAD DAN OVERLOADING METHOD

Tujuan

1. Praktikan mampu memahami konsep static method yang ada di java
2. Mampu membedakan perbedaan method yang menggunakan kata kunci static atau tidak
3. Mampu memahami dan mengimplementasikan method overloading

Ringkasan Materi

A. Overloading Method

Penamaan method pada OOP (Object Oriented Programming) menjadi sangat penting terutama pada pemrograman menggunakan bahasa java. Dalam penamaan method, terkadang tanpa sadar kita memberi nama yang sama pada method yang berbeda sehingga dapat mengakibatkan kesalahan pada saat program dijalankan. Untuk mengatasi hal ini, Java memperkenalkan istilah *overloading*, Overloading adalah teknik penamaan method dengan nama yang sama namun memiliki tipe dan jumlah argumen atau parameter yang berbeda. Sebagai contoh adalah method Hitung pada class Lingkaran, dimana pada class ini terdapat method bernama Hitung dengan parameter a dengan tipe integer.

```
public class Lingkaran{
    public static void Hitung(int a){
        //kode program
    }
}
```

Kemudian pada class tersebut dibuat method baru bernama Hitung namun parameternya bertipe double dengan nama value

```
public static void Hitung(double value){
    //kode program
}
```

Kedua method ini disebut overloading method karena memiliki nama yang sama tetapi tipe dari argumennya berbeda.

B. Overloading Constructor

Sama halnya dengan Overloading Method, Overloading Constructor juga mempunyai karakteristik yang serupa, namun hanya saja peletakannya yang berbeda yaitu pada constructornya saja.

```
public Lingkaran(int alas){
    //kode program
}

public Lingkaran(int alas, int tinggi){
    //kode program
}
```

Pelaksanaan Percobaan

A. Overloading Method

Ketikkan program di bawah ini

1	import java.util.Scanner;
---	---------------------------

```

2 public class Overloading {
3     public static void HitungLuas(int a,int b){
4         int nilai = a*b;
5         System.out.println("maka hasil luas : "+nilai);
6     }
7     public static double HitungLuas(double value, double value2){
8         double nilai = value* value2;
9         return nilai;
10    }
11    public static void main(String[] args) {
12        Scanner in = new Scanner(System.in);
13        System.out.print("masukkan nilai integer 1 : ");
14        int integer1 = in.nextInt();
15        System.out.print("masukkan nilai integer 2 : ");
16        int integer2 = in.nextInt();
17        HitungLuas(integer1, integer2);
18        System.out.print("masukkan nilai double 1 : ");
19        double double1 = in.nextDouble();
20        System.out.print("masukkan nilai double 2 : ");
21        double double2 = in.nextDouble();
22        HitungLuas(integer1, integer2);
23        System.out.println("Maka          hasil          luas          :
24        "+HitungLuas(double1, double2));
25    }
26 }

```

B. Overloading Constructor

Ketikkan program dibawah ini

```

1 public class lingkaran{
2     int alas, tinggi;
3     public lingkaran(int alas){
4         this.alas = alas;
5     }
6     public lingkaran(int alas, int tinggi){
7         this.alas = alas;
8         this.tinggi = tinggi;
9     }
10    public void setAlas(int alas){
11        this.alas = alas;
12    }
13    public void setTinggi(int tinggi){
14        this.tinggi = tinggi;
15    }
16    public int getAlas(){
17        return alas;
18    }
19    public int getTinggi(){
20        return tinggi;
21    }
22    public double hitungLuas(){
23        double hasil = (double) (getTinggi() *getAlas()) /2;
24        return hasil;
25    }

```

26	public void displayMessage() {
27	System.out.println("Hitung Luas : "+hitungLuas());
28	}
29	}

Ketikkan main classnya

1	public class Lingkaran
2	public
3	
4	l.setTinggi(10);
5	l.displayMessage();
6	lingkaran l2 = new lingkaran(4, 10);
7	l2.displayMessage();
8	}
9	}

Data dan Analisis hasil percobaan

A. Overloading Method

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

(Tidak terdapat kesalahan)

The screenshot shows an IDE with a Java file named 'Overloading.java'. The code defines a class 'Overloading' with a 'main' method that demonstrates method overloading. It calls 'hitungLuas' with integer and double parameters. The terminal output shows the program running successfully, displaying the calculated area for both integer and double inputs.

```

1 public class Overloading {
2     public static void main(String[] args) {
3         System.out.print(s:"masukkan nilai double 1 : ");
4         double double1 = in.nextDouble();
5         System.out.print(s:"masukkan nilai double 2 : ");
6         double double2 = in.nextDouble();
7         HitungLuas(integer1, integer2);
8         System.out.println("Maka hasil luas : " + HitungLuas(double1, double2));
9     }
10 }

```

Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\OneDrive\Desktop\modul 3> cmd /C "C:\Users\ASUS\AppData\Roaming\Code\User\globalStorage\pleiades.java-extension-pack-jdk\java\17\bin\java.exe -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:54764 -XX:+ShowCodeDetailsInExceptionMessages -cp "C:\Users\ASUS\AppData\Roaming\Code\User\workspaceStorage\2d731fc33f2592b61494ae22fdfe1171\redhat.java\jdt_ws\modul_3_84a4d5ee\bin" Overloading "

```

masukkan nilai integer 1 : 7
masukkan nilai integer 2 : 7
maka hasil luas : 49
masukkan nilai double 1 : 20
masukkan nilai double 2 : 50
maka hasil luas : 49
Maka hasil luas : 1000.0

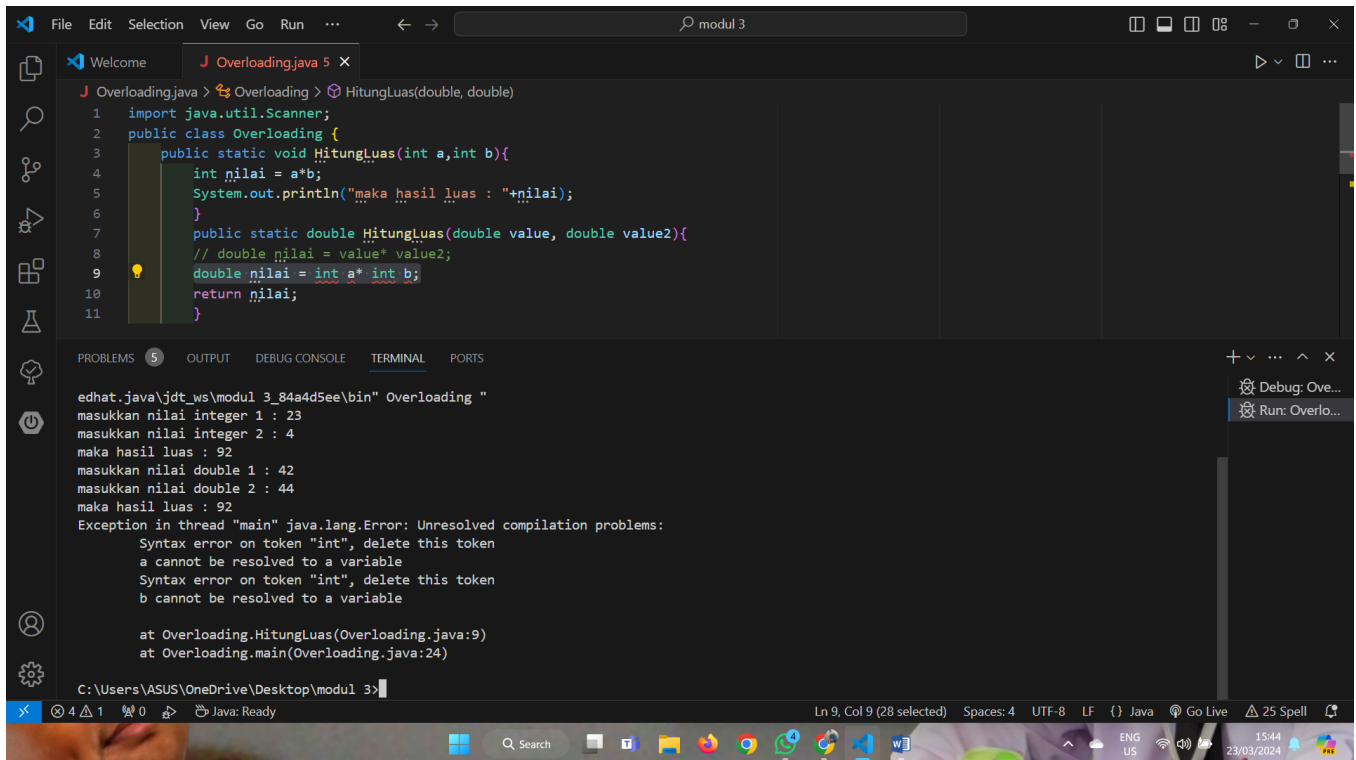
```

C:\Users\ASUS\OneDrive\Desktop\modul 3>

2. Jika pada baris 7, pada parameter double value dan double value2 di hapus dan di ganti menjadi int a dan int b apa yang terjadi? Jelaskan!

Akan terjadi error dikarenakan, jika pada bagian double value dan value2 diganti menjadi int a dan int b, maka saat pemanggilan method dari hitung luas (membutuhkan parameter double) akan mengalami error yaitu kesalahan terhadap penyesuaian tipe data dari double ke int.

Kesalahan ini terjadi karena saat ini hanya ada satu method Hitung Luas yang menerima dua parameter bertipe integer (int a dan int b), sedangkan method Hitung Luas yang menerima dua parameter bertipe double telah dihapus karena perubahan tersebut.



```
File Edit Selection View Go Run ... modul 3
Welcome J Overloading.java 5 X
J Overloading.java > Overloading > HitungLuas(double, double)
1 import java.util.Scanner;
2 public class Overloading {
3     public static void HitungLuas(int a,int b){
4         int nilai = a*b;
5         System.out.println("maka hasil luas : "+nilai);
6     }
7     public static double HitungLuas(double value, double value2){
8         // double nilai = value* value2;
9         double nilai = int a* int b;
10        return nilai;
11    }
12 }

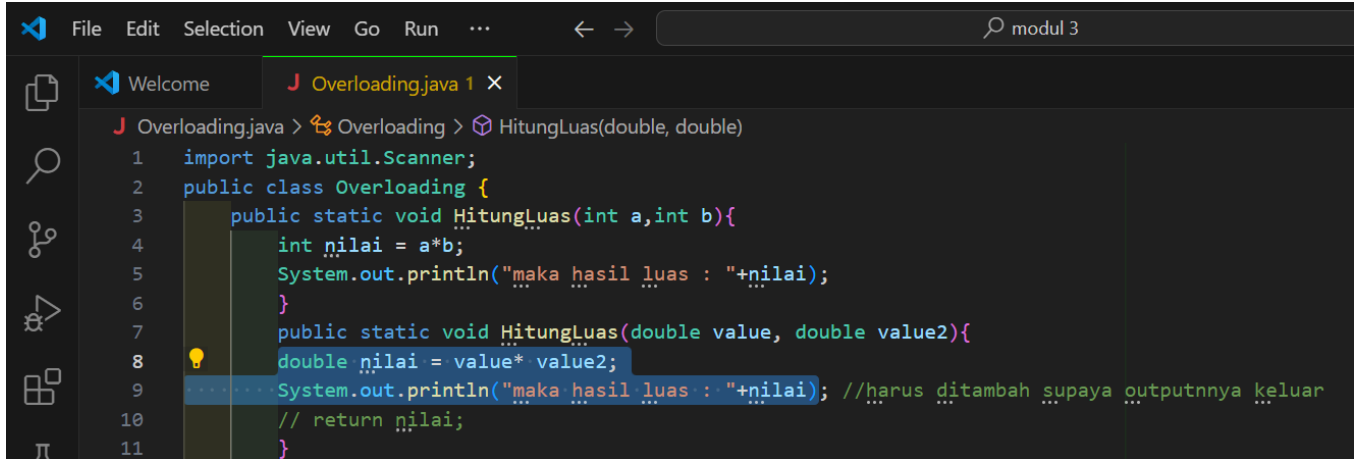
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS
edhat.java\jdt_ws\modul 3_84a4d5ee\bin" Overloading "
masukkan nilai integer 1 : 23
masukkan nilai integer 2 : 4
maka hasil luas : 92
masukkan nilai double 1 : 42
masukkan nilai double 2 : 44
maka hasil luas : 92
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
Syntax error on token "int", delete this token
a cannot be resolved to a variable
Syntax error on token "int", delete this token
b cannot be resolved to a variable

at Overloading.HitungLuas(Overloading.java:9)
at Overloading.main(Overloading.java:24)

C:\Users\ASUS\OneDrive\Desktop\modul 3>
```

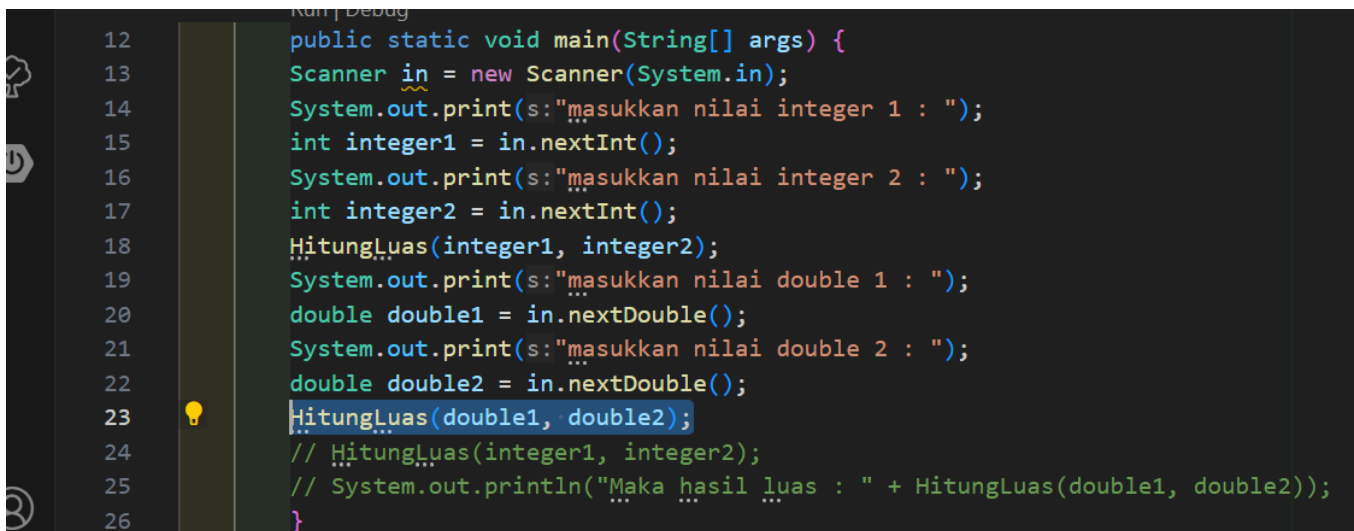
3. Rubah method pada baris ketujuh menjadi method bertipe void, dan lakukan juga perubahan main method.

Jika method `HitungLuas` diubah menjadi bertipe void, ini berarti method tersebut tidak akan memberikan kembali nilai apa pun setelah dieksekusi.



The screenshot shows an IDE window titled 'Overloading.java 1'. The code defines a class 'Overloading' with two static methods named 'HitungLuas'. The first method, on line 3, takes two integers and returns an integer. The second method, on line 7, takes two doubles and returns a double. Line 7 is highlighted with a yellow lightbulb icon, indicating a warning or suggestion. The code is as follows:

```
1 import java.util.Scanner;
2 public class Overloading {
3     public static void HitungLuas(int a,int b){
4         int nilai = a*b;
5         System.out.println("maka hasil luas : "+nilai);
6     }
7     public static void HitungLuas(double value, double value2){
8         double nilai = value* value2;
9         System.out.println("maka hasil luas : "+nilai); //harus ditambah supaya outputnya keluar
10        // return nilai;
11    }
```



The screenshot shows the 'main' method of the 'Overloading' class. It uses a 'Scanner' to take input from the user. It prompts for two integer values and two double values. It then calls the 'HitungLuas' method for each pair of values. The code is as follows:

```
12 public static void main(String[] args) {
13     Scanner in = new Scanner(System.in);
14     System.out.print(s:"masukkan nilai integer 1 : ");
15     int integer1 = in.nextInt();
16     System.out.print(s:"masukkan nilai integer 2 : ");
17     int integer2 = in.nextInt();
18     HitungLuas(integer1, integer2);
19     System.out.print(s:"masukkan nilai double 1 : ");
20     double double1 = in.nextDouble();
21     System.out.print(s:"masukkan nilai double 2 : ");
22     double double2 = in.nextDouble();
23     HitungLuas(double1, double2);
24     // HitungLuas(integer1, integer2);
25     // System.out.println("Maka hasil luas : " + HitungLuas(double1, double2));
26 }
```

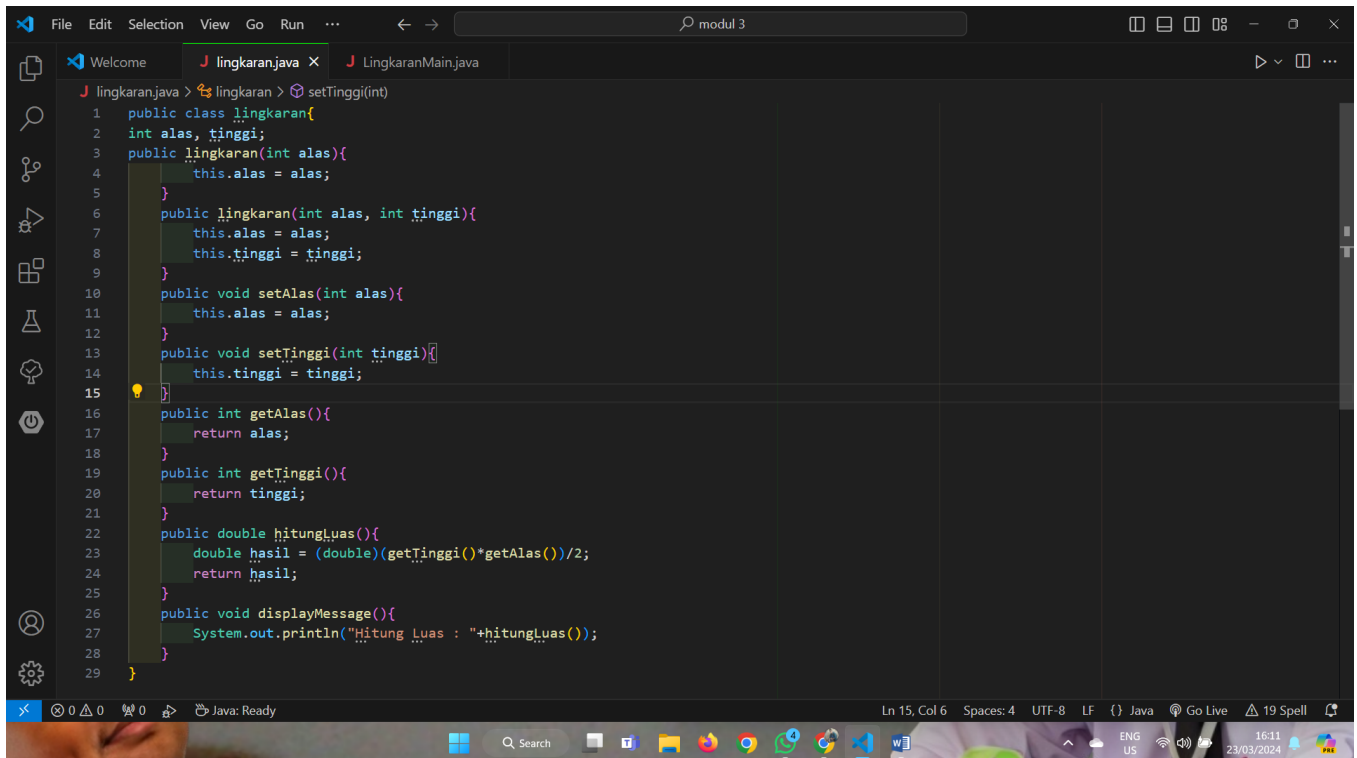
```
c:\Users\ASUS\OneDrive\Desktop\modul
lStorage\pleiades.java-extension-pac
r\workspaceStorage\2d731fc33f2592b61
masukkan nilai integer 1 : 2
masukkan nilai integer 2 : 4
maka hasil luas : 8
masukkan nilai double 1 : 4
masukkan nilai double 2 : 5
maka hasil luas : 20.0
```

B. Overloading Constructor

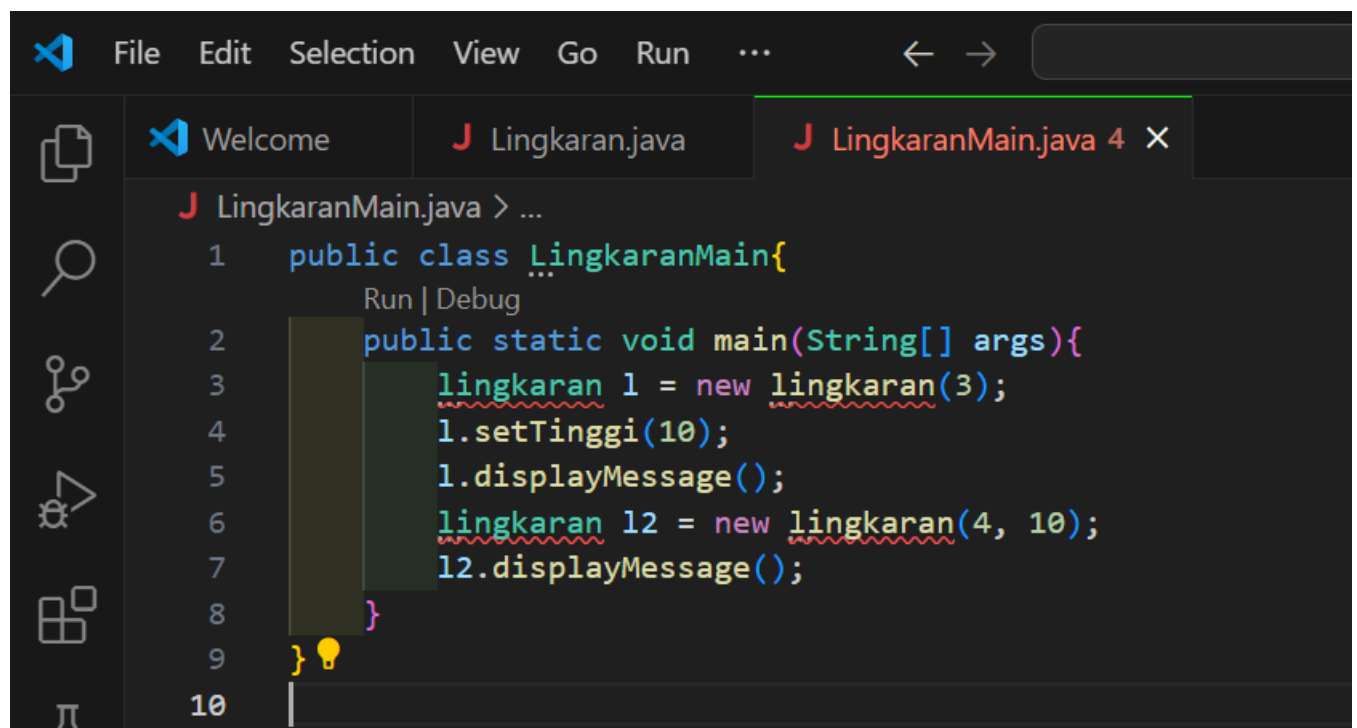
Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

Sebelum :

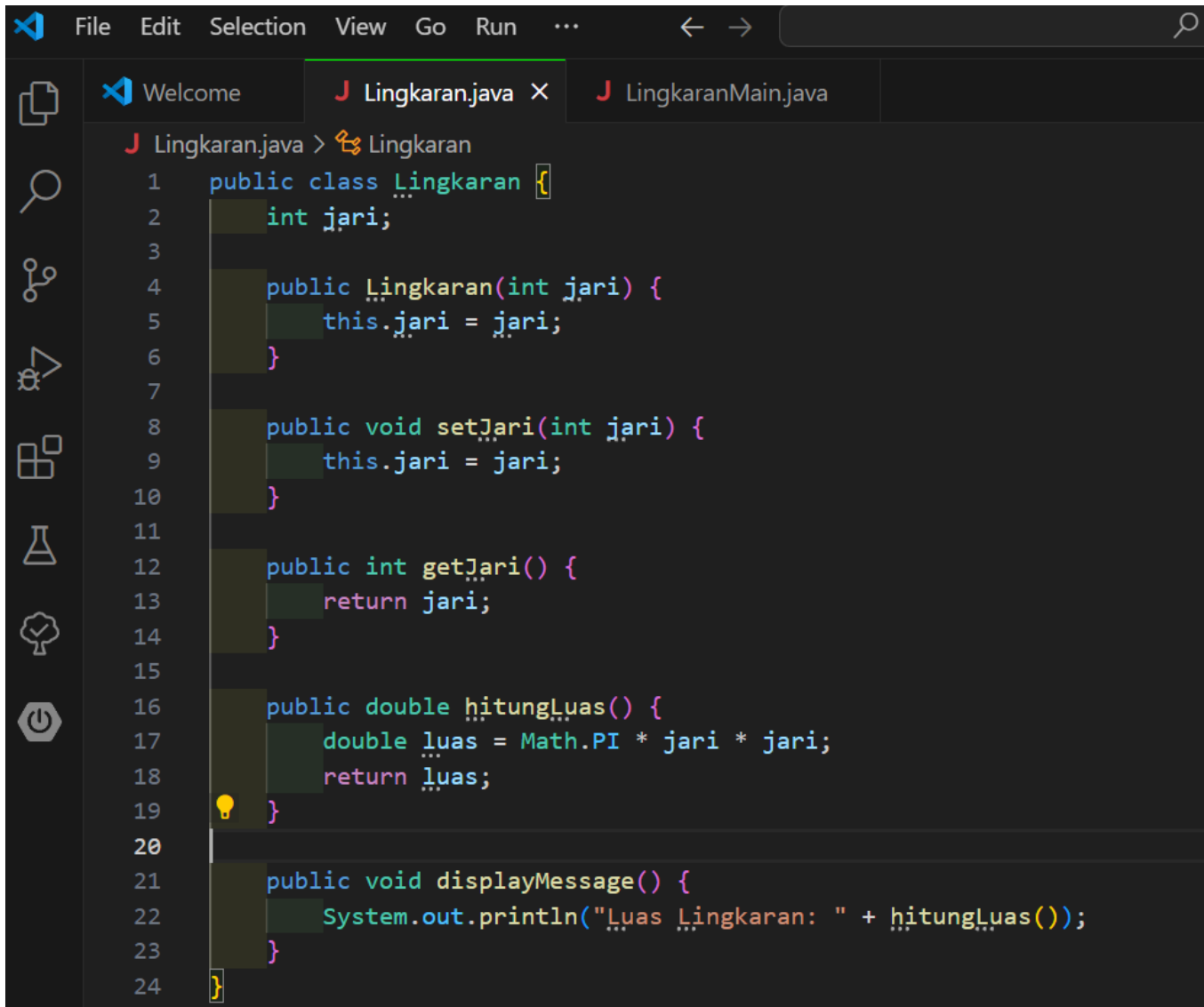


```
1 public class lingkaran{
2     int alas, tinggi;
3     public lingkaran(int alas){
4         this.alas = alas;
5     }
6     public lingkaran(int alas, int tinggi){
7         this.alas = alas;
8         this.tinggi = tinggi;
9     }
10    public void setAlas(int alas){
11        this.alas = alas;
12    }
13    public void setTinggi(int tinggi){
14        this.tinggi = tinggi;
15    }
16    public int getAlas(){
17        return alas;
18    }
19    public int getTinggi(){
20        return tinggi;
21    }
22    public double hitungLuas(){
23        double hasil = (double)(getTinggi()*getAlas())/2;
24        return hasil;
25    }
26    public void displayMessage(){
27        System.out.println("Hitung Luas : "+hitungLuas());
28    }
29 }
```



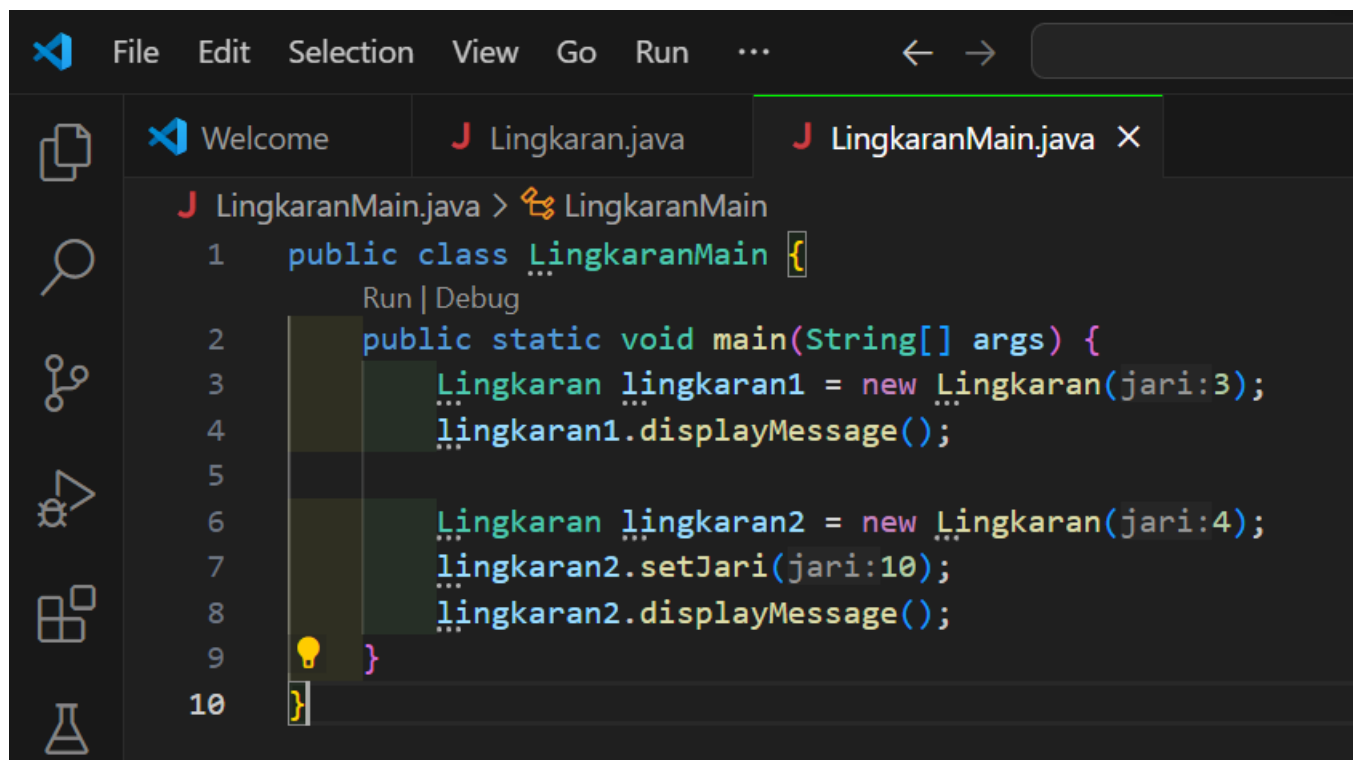
```
1 public class LingkaranMain{
2     public static void main(String[] args){
3         lingkaran l = new lingkaran(3);
4         l.setTinggi(10);
5         l.displayMessage();
6         lingkaran l2 = new lingkaran(4, 10);
7         l2.displayMessage();
8     }
9 }
10
```

Sesudah :



The screenshot shows an IDE with two tabs: 'Welcome' and 'Lingkaran.java'. The 'Lingkaran.java' tab is active, showing the code for the 'Lingkaran' class. The code is as follows:

```
1 public class Lingkaran {
2     int jari;
3
4     public Lingkaran(int jari) {
5         this.jari = jari;
6     }
7
8     public void setJari(int jari) {
9         this.jari = jari;
10    }
11
12    public int getJari() {
13        return jari;
14    }
15
16    public double hitungLuas() {
17        double luas = Math.PI * jari * jari;
18        return luas;
19    }
20
21    public void displayMessage() {
22        System.out.println("Luas Lingkaran: " + hitungLuas());
23    }
24 }
```



The screenshot shows an IDE with two tabs: 'Welcome' and 'LingkaranMain.java'. The 'LingkaranMain.java' tab is active, showing the code for the 'LingkaranMain' class. The code is as follows:

```
1 public class LingkaranMain {
2     Run | Debug
3     public static void main(String[] args) {
4         Lingkaran lingkaran1 = new Lingkaran(jari:3);
5         lingkaran1.displayMessage();
6
7         Lingkaran lingkaran2 = new Lingkaran(jari:4);
8         lingkaran2.setJari(jari:10);
9         lingkaran2.displayMessage();
10    }
11 }
```

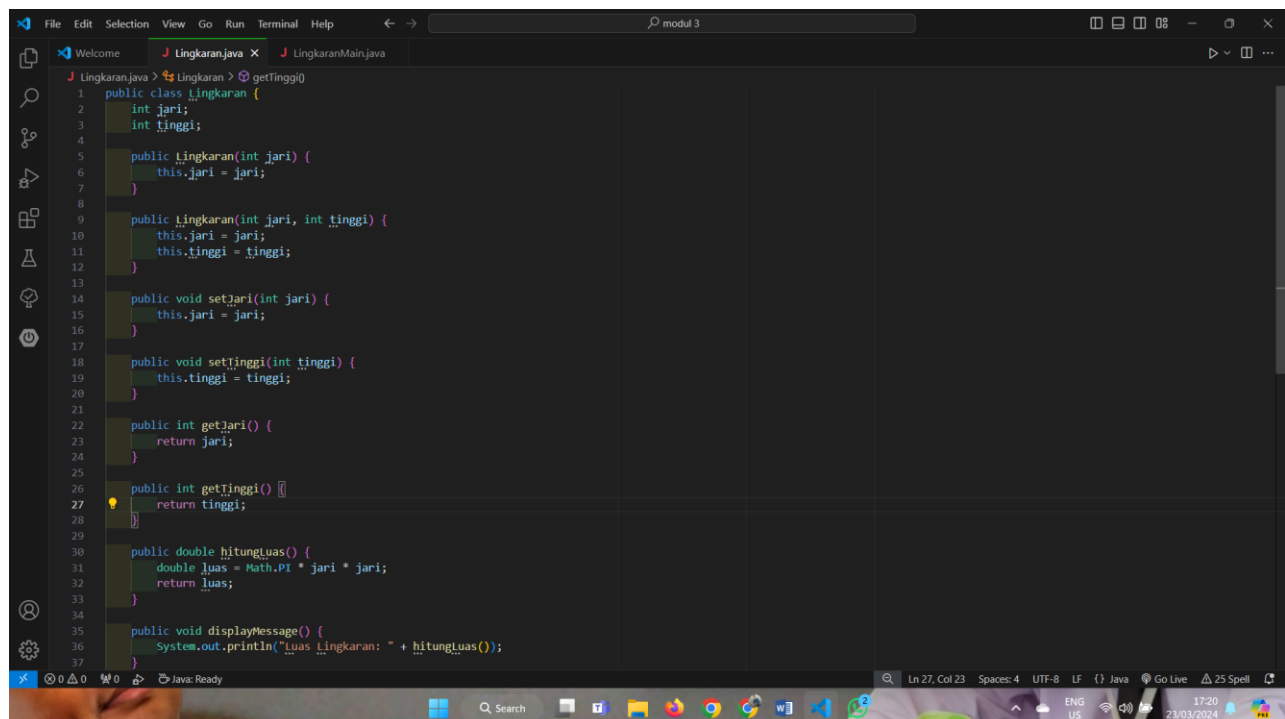
Perubahan yang saya lakukan :

1. Mengubah variabel alas dan tinggi menjadi jari untuk merepresentasikan jari-jari lingkaran.
2. Memperbaiki metode hitungLuas() agar menghitung luas lingkaran menggunakan rumus yang tepat.
3. Mengubah metode displayMessage() untuk mencetak luas lingkaran.
4. Di kelas LingkaranMain, mengubah nama variabel objek dari "l" menjadi lingkaran1 dan dari "12" menjadi lingkaran2 agar lebih deskriptif.
5. Memperbaiki parameter yang dikirim saat membuat objek lingkaran2 agar sesuai dengan konstruktor yang sudah didefinisikan.

2. Pada class lingkaran Tambahkan constructor dengan parameter int tinggi, apa yang terjdadan jelaskan!

Tetap bisa berjalan, namun sebelum penambahan konstruktor dengan parameter int tinggi, kelas Lingkaran hanya dapat diinisialisasi dengan memberikan nilai untuk int jari saja. Ini berarti, saat membuat objek Lingkaran, kita hanya perlu memberikan nilai untuk jari-jari lingkaran.

Namun, setelah penambahan konstruktor baru dengan parameter int tinggi, kelas Lingkaran menjadi lebih fleksibel dengan adanya dua konstruktor yang berbeda. Sekarang, kita dapat membuat objek Lingkaran dengan memberikan nilai untuk jari dan tinggi secara bersamaan saat pembuatan objek. Dengan penambahan konstruktor ini, kita memiliki lebih banyak pilihan dalam inisialisasi objek, karena kita dapat menyertakan nilai tinggi lingkaran bersamaan dengan jari-jari saat membuat objek baru.



```

1 public class Lingkaran {
2     int jari;
3     int tinggi;
4
5     public Lingkaran(int jari) {
6         this.jari = jari;
7     }
8
9     public Lingkaran(int jari, int tinggi) {
10        this.jari = jari;
11        this.tinggi = tinggi;
12    }
13
14    public void setJari(int jari) {
15        this.jari = jari;
16    }
17
18    public void setTinggi(int tinggi) {
19        this.tinggi = tinggi;
20    }
21
22    public int getJari() {
23        return jari;
24    }
25
26    public int getTinggi() {
27        return tinggi;
28    }
29
30    public double hitungLuas() {
31        double luas = Math.PI * jari * jari;
32        return luas;
33    }
34
35    public void displayMessage() {
36        System.out.println("Luas Lingkaran: " + hitungLuas());
37    }
38 }
  
```


3. Pada class lingkaran tambahkan constructor dengan tipe data String alas dan String tinggi, kemudian tambahkan method untuk melakukan parsing atau perubahan tipe data dari String menjadi integer. Setelah itu pada method main lakukan instansiasi objek dengan nama objek Lstring dengan memanggil constructor bertipe data String. Jelaskan!

```
File Edit Selection View Go Run Terminal Help
modul 3

Welcome J Lingkaran.java X J LingkaranMain.java

J Lingkaran.java > Lingkaran
25
26 public class Lingkaran {
27     int alas;
28     int tinggi;
29
30     public Lingkaran(int alas) {
31         this.alas = alas;
32     }
33
34     public Lingkaran(int alas, int tinggi) {
35         this.alas = alas;
36         this.tinggi = tinggi;
37     }
38
39     public Lingkaran(String alas, String tinggi) {
40         this.alas = parseInt(alas);
41         this.tinggi = parseInt(tinggi);
42     }
43
44     private int parseInt(String value) {
45         try {
46             return Integer.parseInt(value);
47         } catch (NumberFormatException e) {
48             System.out.println("Error: " + e.getMessage());
49             return 0;
50         }
51     }
52
53
54     public double hitungLuas() {
55         double luas = (double) (alas * tinggi) / 2;
56         return luas;
57     }
58
59     public void displayMessage() {
60         System.out.println("Luas Lingkaran: " + hitungLuas());
61     }
62 }

Java: Ready
Ln 52, Col 1 Spaces: 4 UTF-8 LF {} Java Go Live 30 Spell

File Edit Selection View Go Run Terminal Help
modul 3

Welcome J Lingkaran.java X J LingkaranMain.java

J LingkaranMain.java > LingkaranMain
1 // public class LingkaranMain {
2 //     public static void main(String[] args) {
3 //         Lingkaran lingkaran1 = new Lingkaran(3);
4 //         lingkaran1.displayMessage();
5 //     }
6 //     Lingkaran lingkaran2 = new Lingkaran(4);
7 //     lingkaran2.setTinggi(10);
8 //     lingkaran2.displayMessage();
9 // }
10 // }
11
12 public class LingkaranMain {
13     String[] args = LingkaranMain.main(String[])
14     public static void main(String[] args) {
15         Lingkaran l1 = new Lingkaran(alas:3); // contoh penggunaan konstruktor dengan parameter int alas saja
16         l1.displayMessage();
17
18         Lingkaran l2 = new Lingkaran(alas:"4", tinggi:"10"); // contoh penggunaan konstruktor dengan parameter String alas dan tinggi
19         l2.displayMessage();
20     }
21 }
```

Perubahan utama antara kode sebelumnya dan hasil akhir adalah penambahan konstruktor baru yang menerima parameter dalam bentuk String untuk alas dan tinggi, serta penambahan metode untuk mengonversi nilai dari String menjadi int. Dengan penambahan ini, kita dapat membuat objek Lingkaran dengan memberikan nilai dalam format String, yang kemudian akan diubah menjadi int di dalam konstruktor.

Perbedaan tersebut meliputi:

1. Penambahan Konstruktor Baru:

- Sebelumnya, hanya terdapat dua konstruktor: satu untuk alas bertipe int dan satu untuk alas dan tinggi bertipe int.
- Dalam hasil akhir, konstruktor baru ditambahkan dengan dua parameter String untuk alas dan tinggi.

2. Penambahan Metode `parseToInt()`:

- Dalam hasil akhir, metode baru `parseToInt(String value)` digunakan untuk mengonversi nilai dari String menjadi int.

3. Penggunaan Objek dengan Parameter String:

- Di dalam metode `main()`, kita menunjukkan cara membuat objek Lingkaran dengan memberikan nilai dalam format String. Nilai ini kemudian akan diubah menjadi int di dalam konstruktor.

Tugas Praktikum

Melanjutkan dari bab 1 dan 2, buatlah method berikut :

1. Method `bacaFile("pathFile")`
Fungsinya adalah memberi nilai pada kelas kalian dengan nilai atribut sesuai yang ada di file txt. Apabila kelas kalian pada bab 1 memiliki atribut judul dan penulis, maka pada file txt akan ada 2 nilai untuk judul dan nilai penulis (isi txt : DILAN 1990; Pidi Baiq)
2. Method `simpanFile("namaFile")`
Fungsinya adalah menyimpan nilai atribut kelas.
3. Method `hitungRoyalti` dengan parameter harga 1 buku, yang mengembalikan nilai 10% dari penjualan bulan itu. (1 parameter)
4. Method `hitungRoyalti` dengan parameter harga 1 buku dan nilai persen yang ingin diambil dari penjualan bulan itu. (2 parameter)