

Academic Year 2021/2022

# Pemrograman Berorientasi Objek

Rofilde Hasudungan

Lecturer

Department of Informatic

Universitas Muhammadiyah Kalimantan Timur

October 1, 2021

# Outline

- 1 Pengenalan
- 2 Mendefinisikan Method
- 3 Menggunakan Method
- 4 Parameter pada Method
- 5 Void Method
- 6 Return Value Method
- 7 Method Overloading

# Pengenalan

Secara sederhana method dapat diartikan group kode yang memiliki operasi dengan tujuan spesifik. Method merupakan salah satu cara untuk menghilangkan redudansi kode serta memungkinkan penggunaan kembali, serta dapat digunakan untuk memodularisasi kode dan meningkatkan kualitas dari program.

# Mendefinisikan Method

Suatu method dapat didefinisikan dengan menggunakan format berikut:

```
1 modifier returnType methodName(list of parameters) {  
2     // Method body;  
3 }
```

- 1 modifier dapat berisi default (tidak ada), public, private, dan protected (dijelaskan kemudian)
- 2 returnValue, tipe data dari nilai yang mungkin dihasilkan oleh method (dapat berupa int, long, double, atau objek) atau void jika tidak menghasilkan nilai.
- 3 methodName ialah nama dari method, dimana memiliki format jika satu suku kata ditulis normal, namun jika lebih, maka huruf pertama dari suku kata selanjutnya haruslah kapital, seperti hitungLuas(), hasilBagi(), dst.
- 4 method body, berisi operasi/statement yang dijalankan oleh method ketika dipanggil (digunakan).

# Mendefinisikan Method

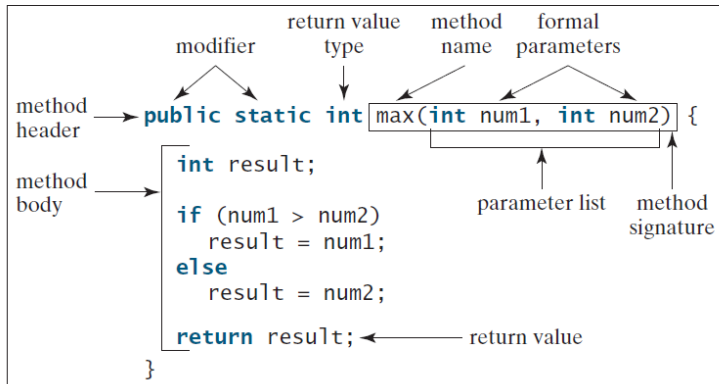
Suatu method dapat didefinisikan dengan menggunakan format berikut:

```
1  modifier returnType methodName(list of parameters) {  
2      // Method body;  
3  }
```

- 1 modifier dapat berisi default (tidak ada), public, private, dan protected (dijelaskan kemudian), dan/atau static
- 2 returnValue, tipe data dari nilai yang mungkin dihasilkan oleh method (dapat berupa int, long, double, atau objek) atau void jika tidak menghasilkan nilai.
- 3 methodName ialah nama dari method, dimana memiliki format jika satu suku kata ditulis normal, namun jika lebih, maka huruf pertama dari suku kata selanjutnya haruslah kapital, seperti hitungLuas(), hasilBagi(), dst.
- 4 method body, berisi operasi/statement yang dijalankan oleh method ketika dipanggil (digunakan).

# Mendefinisikan dan Menggunakan Method

## Define a method



## Invoke a method

```
int z = max(x, y);
```

↑ ↑  
actual parameters  
(arguments)

# Contoh

```
1 public class TestMaxMinEqual{
2     public static void main(String[] args) {
3         int a=5, b = 7, c =7;
4         System.out.printf("max(%d, %d) = %d", a, b, TestMaxMinEqual.max(a,
5             b));
6         System.out.printf("max(%d, %d) = %d", a, b, TestMaxMinEqual.min(a,
7             b));
8         System.out.printf("max(%d, %d) = %d", c, b, TestMaxMinEqual.max(c,
9             b));
10    }
11
12    public static int max(int a, int b) {
13        if (a > b) {
14            return a;
15        } else {
16            return b;
17        }
18    }
19 }
```

```
17 public static int min(int a, int b) {
18     if (a < b) {
19         return a;
20     } else {
21         return b;
22     }
23 }
24
25 public static boolean equal(int a, int b) {
26     if (a == b) {
27         return true;
28     } else {
29         return false;
30     }
31 }
32 }
```

---



# Parameter Method

Suatu method dapat memiliki satu/lebih parameter, dan bahkan tidak sama sekali. Parameter digunakan untuk memberikan nilai kepada body method untuk diolah/diproses.

---

```
1 \begin{lstlisting}[language=java, caption={}, label={}]
2 modifier returnType methodName(data-type par1, data-type par2,
   ... ) {
3     // Method body;
4 }
```

---

Seperti pada format di atas,

- 1 parameter didefinisikan seperti variabel
- 2 parameter tidak dapat didefinisikan beserta nilai tertentu

# Contoh Method dan Parameter

```
1 public class HitungLingkaran {
2     public static void main(String[] args) {
3         System.out.println("Luas Lingkaran dengan radius 7 ialah " +
4             luasLingkaran(7, 22/7));
5         System.out.println("Luas Lingkaran dengan radius 8 ialah " +
6             luasLingkaran(8, 3.14));
7     }
8
9     public static double luasLingkaran(double radius, double phi) {
10         double luas;
11         luas = Math.pow(radius, 2) * phi;
12         return luas;
13     }
14 }
```

# Void Method

Void method ialah method yang tidak memiliki nilai balik, sehingga method tersebut tidak memiliki nilai. Method seperti ini umumnya digunakan ketika tidak ada proses lanjutan dari hasil method tersebut.

---

```
1 public class ExVoidMethod {
2     public static void main(String[] args) {
3         luasSegitiga(7, 8);
4     }
5     public static void luasSegitiga(double alas, double tinggi) {
6         double luas;
7         luas = alas * tinggi / 2;
8         System.out.print("Luas Segitiga dengan ");
9         System.out.print("Alas =" + alas);
10        System.out.print(" dan Tinggi =" + tinggi);
11        System.out.print(" adalah =" + luas);
12    }
13 }
```

---

# Void Method

Perhatikan contoh pada halaman sebelumnya, luas segitiga yang dihasilkan langsung di-output-kan dan method segitiga tidak memiliki tipe data artinya tidak memiliki nilai. Jika kita ingin mengolah nilai hasil perhitungan, misalkan kita ingin ubah satuannya, maka tidak dimungkinkan.

# Method return value

Method ini memiliki nilai balik atau dengan kata lain method ini memiliki nilai sehingga method ini didefinisikan dengan tipe data tertentu sesuai dengan nilainya.

---

```
1 \begin{lstlisting}[language=java, caption={}, label={}]  
2 modifier returnType methodName(data-type par1, data-type par2,  
   ... ) {  
3     // Method body;  
4     // return VALUE  
5 }
```

---

Format pembuatan method ini sama dengan void method, bedanya

- ① returnType dapat diisi berbagai tipe data atau objek
- ② pada akhir baris kode terdapat nilai balik sesuai dengan tipe data pada (1)

# Contoh

```
1 public class ExReturnValueMethod {
2     public static void main(String[] args) {
3         System.out.println("Luas Segitiga = " + luasSegitiga(6, 7) + "
4             meter");
5         System.out.println("Luas Segitiga = " + luasSegitiga(6, 7)*100 + "
6             centi meter");
7     }
8
9     public static double luasSegitiga(double alas, double tinggi) {
10         double luas;
11         luas = alas * tinggi / 2;
12         return luas;
13     }
14 }
```

# Method Overloading

Bahasa pemrograman java memungkinkan kita untuk membuat method dengan nama yang sama, namun memiliki jumlah parameter yang berbeda. Hal semacam ini disebut dengan method overloading. Penggunaan method overloading dalam pengembangan aplikasi sangatlah banyak. Dengan mengimplementasikan method overloading, suatu method seperti memiliki parameter yang dinamis.

# Contoh Method Overloading

```
1 public class HitungLingkaran {
2     public static void main(String[] args) {
3         System.out.println("Luas Lingkaran dengan radius 7 ialah " +
4             luasLingkaran(7));
5         System.out.println("Luas Lingkaran dengan radius 8 ialah " +
6             luasLingkaran(7, 3.14));
7     }
8     public static double luasLingkaran(double radius, double phi) {
9         double luas;
10        luas = Math.pow(radius, 2) * phi;
11        return luas;
12    }
13    public static double luasLingkaran(double radius) {
14        double luas;
15        luas = Math.pow(radius, 2) * 22/7;
16        return luas;
17    }
18 }
```