

## Pertemuan 5 (Asinkron)

# Void Method & Non Void Method

Galang P Mahardhika

## Pengantar Materi

Pada pertemuan ini kita akan mempelajari apa itu method dan bagaimana cara pemanfaatan method untuk membantu kita dalam membuat program. Pertama-tama kita akan mengenal dan memahami cara kerja method, lalu kita akan mencoba untuk membuat method. Setelah berhasil membuat method, kemudian kita akan mencoba menggunakan method tersebut dalam program yang kita buat.

## Topik Materi

1. Method
2. Struktur Method
3. Latihan Mandiri

## 1. Method

### 1.1. Definisi Method

Secara sederhana, **method** adalah suatu kelompok/blok program yang menyimpan kumpulan perintah untuk melakukan tugas tertentu. Method (kumpulan perintah) dapat dijalankan/dipanggil dengan prosedur pemanggilan method (akan kita pelajari pada pertemuan ini). Pada beberapa konsep pemrograman, method juga dikenal dengan nama *function*/fungsi (w3school.com). Dari sudut pandang fungsionalitas, method merupakan salah satu sarana bagi programmer untuk memecah program menjadi bagian kecil agar dapat digunakan secara berulang.

Selain dapat melakukan tugas tertentu, sebuah method juga dapat mengembalikan suatu nilai ketika dipanggil (www. geeksforgeeks.org). Method yang mengembalikan suatu nilai umumnya dikenal dengan istilah **Non Void Method** atau **Return Method**, sedangkan method yang tidak mengembalikan suatu nilai umumnya dikenal dengan istilah **Void Method**. Mungkin Anda akan bertanya, apa yang dimaksud dengan nilai balik itu? Jangan khawatir, nanti kita akan mempelajari hal ini dengan menggunakan contoh praktiknya.

### 1.2. Main Method (Method Utama)

Selain Non Void Method dan Void Method, dalam konsep pemrograman kita juga sering mendengar istilah Main Method atau Method Utama. **Main Method** merupakan blok program yang akan dieksekusi pertama kali ketika aplikasi dijalankan. Dalam hal ini Main Method berperan sebagai driver dan wajib ada agar aplikasi dapat berjalan. Setiap aplikasi (kita kenal sebagai project) hanya boleh memiliki satu main method. Main method dideklarasikan sebagai berikut:

```
public static void main(String[] args) {  
    <pernyataan/ekspresi>  
}
```

Method tersebut tentu saja tidak asing bukan? Sejauh ini kita telah belajar membuat program dalam satu Main Method. Apakah bisa kita menuliskan keseluruhan program yang kita buat pada bagian Main Method? Tentu saja bisa, tapi ketika program yang kita buat cukup kompleks, hal ini tentu saja tidak efektif dan efisien untuk dilakukan.

## 2. Struktur Method

### 2.1. Deklarasi Method

Tiap method memiliki bagian-bagian penyusun yang harus didefinisikan saat method tersebut dideklarasikan. Bagian tersebut adalah **Tipe Nilai Balik/Return Type**, **Nama Method**, **Parameter** (opsional), **Isi Method/Method Body**, dan **Nilai Balik/Return Value** (opsional). Berikut adalah struktur yang digunakan untuk mendeklarasikan suatu method:

```
static <tipeNilai> <nama> (<parameter>){  
    <isiMethod>  
    <nilaiBalik>  
}
```

- static** : Kata kunci “static” hanya digunakan jika kita membuat method untuk dipanggil pada Main Method. Kedepannya kita akan membuat method pada Kelas yang berbeda. Cara tersebut akan kita pelajari pada pertemuan-pertemuan selanjutnya.
- <tipeNilai>** : Bagian ini berisi tipe dari nilai yang akan dikembalikan. Saat ini, tipe nilai yang kita ketahui adalah tipe nilai yang sama seperti tipe data pada variabel, yaitu **int**, **double**, **char**, **String**, dan lain-lain. Untuk method yang tidak mengembalikan nilai (Void Method) kita dapat menuliskan kata kunci “**void**”.
- <nama>** : Ini adalah bagian untuk mendeklarasikan nama method. Aturan penamaan sama dengan aturan penamaan untuk variabel. Jangan lupa untuk membuat nama yang unik, spesifik dan merepresentasikan fungsi dari method.
- <parameter>** : Parameter merupakan variabel yang menampung nilai untuk diproses pada bagian Isi Method. Nilai yang diberikan para parameter disebut dengan argumen. Parameter bersifat opsional (bisa ada atau tidak ada). Cara mendeklarasikan suatu parameter sama dengan cara mendeklarasikan suatu variabel (**<tipeData> <namaParameter>**). Satu hal yang perlu diperhatikan adalah tanda kurung buka dan kurung tutup harus tetap kita gunakan meskipun kita tidak menggunakan suatu parameter.

- <isiMethod>** : Bagian Isi Method adalah segala proses yang terjadi pada suatu method. Bagian ini dapat berisi deklarasi variabel atau pernyataan.
- <nilaiBalik>** : Nilai balik merupakan nilai yang dikirimkan oleh suatu method. Nilai balik hanya digunakan pada jenis Non Void Method sehingga sifatnya opsional. Satu hal yang perlu diperhatikan adalah tipe data dari nilai yang dikembalikan harus sesuai dengan tipe nilai balik yang didefinisikan saat mendeklarasikan suatu method. Untuk mengembalikan suatu nilai kita perlu menggunakan kata kunci **return**.

Berikut adalah contoh Void Method. Seperti dijelaskan sebelumnya, Void Method merupakan void yang tidak mengembalikan nilai apapun, sehingga kita tidak perlu menggunakan kata kunci return di jenis method ini.

#### 2.1.1. Contoh Void Method Tanpa Parameter

```
package p5a;

public class P5a {

    static void sapa(){
        System.out.println("Halo Dunia");
    }

    public static void main(String[] args) {
        // TODO code application logic here
    }
}
```

#### 2.1.2. Contoh Void Method Dengan Parameter

```
package p5a;

public class P5a {

    static void tampilLuas(int panjang, int lebar){
        System.out.print("Luas : ");
        System.out.println(panjang * lebar);
    }

    public static void main(String[] args) {
        // TODO code application logic here
    }
}
```

Berikut adalah contoh Non Void Method. Jenis method ini mengembalikan nilai balik, nilai yang dikembalikan diletakkan sesudah kata kunci **return**. Perhatikan juga pada bagian tipe nilai baliknya. Dapat dilihat bahwa tipe nilai balik harus sesuai dengan nilai yang akan dikembalikan.

### 2.1.3. Contoh Non Void Method Tanpa Parameter

```
package p5a;

public class P5a {

    static String password(){
        return "123456";
    }

    public static void main(String[] args) {
        // TODO code application logic here
    }
}
```

### 2.1.4. Contoh Non Void Method Dengan Parameter

```
package p5a;

public class P5a {

    static double hitungLuas(double alas, double tinggi){
        return (0.5 * alas * tinggi);
    }

    public static void main(String[] args) {
        // TODO code application logic here
    }
}
```

## 2.2. Pemanggilan Method

Method dapat dipanggil dengan menuliskan nama methodnya kemudian diikuti dengan memberikan argumen (jika menggunakan parameter). Perlu diperhatikan, apabila suatu method tidak memiliki parameter, kita tetap harus menuliskan tanda kurung buka dan kurung tutup pada method yang ingin dipanggil. Baik Void Method maupun Non Void Method cara pemanggilannya relatif sama, hanya saja yang membedakan adalah cara penanganannya. Non Void Method merupakan method yang memberikan nilai balik, sehingga method ini akan menghasilkan nilai yang dapat dioperasikan (disimpan

pada sebuah variabel, ditampilkan, dijumlahkan, ditampilkan, dan lain-lain). Berikut adalah contoh pemanggilan Void Method.

### 2.2.1. Contoh Pemanggilan Void Method

```
package p5a;

public class P5a {

    static void sapa(){
        System.out.println("Halo Dunia");
    }

    static void tampilLuas(int panjang, int lebar){
        System.out.print("Luas : ");
        System.out.println(panjang * lebar);
    }

    public static void main(String[] args) {
        sapa();
        tampilLuas(3, 5);    //3 dan 5 adalah argumen
    }
}
```

Method **sapa()** merupakan method tanpa parameter, sehingga untuk menggunakan method tersebut kita cukup menuliskan nama method disertai dengan tanda kurung buka dan kurung tutup saja. Sedangkan **tampilLuas()** merupakan method dengan parameter, sehingga untuk menggunakan method tersebut kita harus memasukkan argumen. Tipe data untuk argumen yang diberikan harus sesuai dengan tipe data parameternya. Pada contoh tersebut, argumen yang dimasukkan adalah 3 dan 5 yang merupakan bilangan bulat. Berikutnya adalah contoh pemanggilan Non Void Method. Pemanggilan dan penanganan Non Void Method dapat dilihat pada contoh berikut:

### 2.2.2. Contoh Pemanggilan Non Void Method

```
package p5a;

public class P5a {

    static String password(){
        return "123456";
    }

    static double hitungLuas(double alas, double tinggi){
        return (0.5 * alas * tinggi);
    }
}
```

```
}  
  
public static void main(String[] args) {  
    String pass = password();  
    System.out.println(pass);  
  
    System.out.print("Luas Segitiga: ");  
    System.out.println(hitungLuas(3, 5));    //3 dan 5 adalah argumen  
}  
}
```

Method **password()** merupakan method tanpa parameter. Untuk menggunakannya kita cukup menuliskan **password();** saja. Namun method **password()** ini mengembalikan nilai balik berupa String, sehingga kita harus menangani nilai balik tersebut. Pada contoh tersebut, nilai balik saya simpan dalam sebuah variabel bernama “pass” yang kemudian saya tampilkan hasilnya. Penggunaan method **hitungLuas()** dilakukan dengan memberikan argumen sesuai dengan tipe parameternya. Pada contoh tersebut, saya coba untuk langsung menampilkan nilai baliknya dengan **System.out.println()**.

### 3. Latihan Mandiri

Latihan mandiri digunakan untuk menguji pemahaman Anda terkait materi yang telah Anda pelajari pada pertemuan ini. Jawaban dari latihan mandiri disertakan pada bagian akhir. Untuk menguji kemampuan Anda, silakan kerjakan latihan mandiri tanpa melihat jawaban terlebih dahulu. Jika memang Anda merasa kesulitan silakan melihat jawaban. Jawaban yang ada bisa jadi berbeda dengan kode yang telah Anda kerjakan, karena ada banyak solusi untuk menyelesaikan suatu kasus dalam pemrograman. Selamat mengerjakan.

#### 3.1. Latihan Pertama

Buatlah sebuah method untuk menampilkan teks dengan format “Halo, <panggilan> <nama>”, contoh “Halo, Mbak Alima”. Terdapat 2 parameter dalam method tersebut, yaitu “panggilan” dan “nama”. Parameter “panggilan” dapat menerima argumen kode pemanggilan dengan aturan sebagai berikut:

Kode = 1, maka menampilkan panggilan “Mas”  
Kode = 2, maka menampilkan panggilan “Mbak”  
Kode = 3, maka menampilkan panggilan “Pak”  
Kode = 4, maka menampilkan panggilan “Bu”

Panggilan tersebut kemudian akan digabungkan dengan argumen kedua yaitu “nama” (bertipe String). Method tersebut kemudian dapat digunakan pada Main Method sebagai berikut (perhatikan argumen yang digunakan):

```
public static void main(String[] args) {
    menyapa(1, "Andi");
    menyapa(2, "Bety");
    menyapa(3, "Ciko");
    menyapa(4, "Dini");
}
```

Jika dijalankan, maka akan menampilkan keluaran sebagai berikut (perhatikan bahwa argumen yang digunakan telah terkonversi menjadi teks “Mas”, “Mbak”, “Pak”, dan “Bu”).

```
run:
Halo, Mas Andi
Halo, Mbak Bety
Halo, Pak Ciko
Halo, Bu Dini
```

### 3.2. Latihan Kedua

Sebuah Warteg (Warung Tegal) yang baru buka ingin memberikan promo makanan gratis kepada para pelanggannya. Para pelanggan diminta untuk memilih 3 menu (secara bebas) dengan menggunakan sebuah aplikasi. Menu yang dapat dipilih adalah sebagai berikut:

1. Nasi Telur
2. Nasi Ayam
3. Nasi Lele
4. Nasi Nila

Untuk memilih sebuah menu, pelanggan cukup mengetikkan nomor dari menu yang ingin dipilih. Setelah memilih ketiga menu, maka akan muncul keterangan menu-menu apa saja yang telah dipilih. Jika nomor yang dimasukkan tidak sesuai (bukan 1 sampai 4), maka informasi yang muncul pada pilihan menu adalah **[Salah Pilih]**. Buat penyelesaian kasus tersebut dengan pendekatan method sehingga dapat digunakan pada Main Method sebagai berikut:

```
public static void main(String[] args) {
    String pilihan1 = bacaMenu();
    String pilihan2 = bacaMenu();
    String pilihan3 = bacaMenu();

    System.out.print("Anda memesan " + pilihan1 + ", ");
    System.out.println(pilihan2 + ", dan " + pilihan3);
}
```



Jika dijalankan dan diberi masukan 4, 1, dan 2, maka akan muncul keluaran seperti gambar berikut (sebelah kiri). Kemudian jika diberikan masukan 3, 5, dan 2, maka akan muncul keluaran seperti gambar berikut (sebelah kanan).

```
run:
Menu pilihan:
1. Nasi Telur
2. Nasi Ayam
3. Nasi Lele
4. Nasi Nila
Masukkan menu yang dipilih (1-4): 4

Menu pilihan:
1. Nasi Telur
2. Nasi Ayam
3. Nasi Lele
4. Nasi Nila
Masukkan menu yang dipilih (1-4): 1

Menu pilihan:
1. Nasi Telur
2. Nasi Ayam
3. Nasi Lele
4. Nasi Nila
Masukkan menu yang dipilih (1-4): 2

Anda memesan Nasi Nila, Nasi Telur, dan Nasi Ayam
```

```
run:
Menu pilihan:
1. Nasi Telur
2. Nasi Ayam
3. Nasi Lele
4. Nasi Nila
Masukkan menu yang dipilih (1-4): 3

Menu pilihan:
1. Nasi Telur
2. Nasi Ayam
3. Nasi Lele
4. Nasi Nila
Masukkan menu yang dipilih (1-4): 5

Menu pilihan:
1. Nasi Telur
2. Nasi Ayam
3. Nasi Lele
4. Nasi Nila
Masukkan menu yang dipilih (1-4): 2

Anda memesan Nasi Lele, [Salah Pilih], dan Nasi Ayam
```

### 3.3. Latihan Ketiga

Saat ini kita telah mempelajari proses pembacaan input (masukan) dengan fitur Scanner. Sebelum membaca suatu nilai (misal membaca nilai String), biasanya kita menyisipkan teks seperti “Masukkan nama Anda :”, “Masukkan NIM Anda :”, “Masukkan pilihan Anda:”, dan lain sebagainya. Jika dilakukan berkali-kali, tentu saja hal tersebut menjadi tidak efisien. Agar terlihat lebih singkat, buatlah sebuah method dengan nilai balik bertipe String untuk kasus tersebut. Method ini memiliki sebuah parameter yang digunakan untuk menampilkan teks sisipan yang dimunculkan sebelum proses pembacaan nilai String. Kemudian buat juga satu method yang dapat digunakan untuk menampilkan rangkaian teks dengan argumen berupa nama, nim, dan email. Kedua method yang dibuat dapat dipanggil pada Main Method dengan cara berikut:

```
public static void main(String[] args) {
    String nama = bacaTeks("Nama : ");
    String nim = bacaTeks("NIM : ");
    String email = bacaTeks("Email : ");

    tampilInfo(nama, nim, email);
}
```



Jika dijalankan, dan diberikan masukan pertama “Galang Prihadi Mahardhika”, lalu masukan kedua “12345678”, lalu masukan ketiga “galang.prihadi@uii.ac.id”, maka aplikasi tersebut akan menampilkan keluaran seperti gambar berikut:

```
run:
Nama   : Galang Prihadi Mahardhika
NIM    : 12345678
Email  : galang.prihadi@uii.ac.id

Halo, Galang Prihadi Mahardhika (12345678)
Informasi telah dikirim ke email galang.prihadi@uii.ac.id
```

## Jawaban Latihan Pertama (Sesuaikan nama Package dan nama Class dengan Project yang Anda buat)

```
package p5a;

import java.util.Scanner;

public class P5a {

    static void menyapa(int kode, String nama){
        switch (kode){
            case 1 : System.out.print("Halo, Mas "); break;
            case 2 : System.out.print("Halo, Mbak "); break;
            case 3 : System.out.print("Halo, Pak "); break;
            case 4 : System.out.print("Halo, Bu "); break;
        }

        System.out.println(nama);
    }

    public static void main(String[] args) {
        menyapa(1, "Andi");
        menyapa(2, "Bety");
        menyapa(3, "Ciko");
        menyapa(4, "Dini");
    }
}
```

## Jawaban Latihan Kedua (Sesuaikan nama Package dan nama Class dengan Project yang Anda buat)

```
package p5a;

import java.util.Scanner;

public class P5a {

    static String bacaMenu(){
        Scanner baca = new Scanner(System.in);

        System.out.println("Menu pilihan:");
        System.out.println("1. Nasi Telur");
        System.out.println("2. Nasi Ayam");
        System.out.println("3. Nasi Lele");
        System.out.println("4. Nasi Nila");

        System.out.print("Masukkan menu yang dipilih (1-4): ");
        int pilihan = baca.nextInt();
        System.out.println();

        String menu;
        switch (pilihan){
            case 1 : menu = "Nasi Telur"; break;
            case 2 : menu = "Nasi Ayam"; break;
            case 3 : menu = "Nasi Lele"; break;
            case 4 : menu = "Nasi Nila"; break;
            default : return "[Salah Pilih]";
        }

        return menu;
    }

    public static void main(String[] args) {
        String pilihan1 = bacaMenu();
        String pilihan2 = bacaMenu();
        String pilihan3 = bacaMenu();

        System.out.print("Anda memesan " + pilihan1 + ", ");
        System.out.println(pilihan2 + ", dan " + pilihan3);
    }
}
```

**Jawaban Latihan Ketiga** (Sesuaikan nama Package dan nama Class dengan Project yang Anda buat)

```
package p5a;

import java.util.Scanner;

public class P5a {

    static String bacaTeks(String pesan){
        Scanner baca = new Scanner(System.in);
        System.out.print(pesan);
        return baca.nextLine();
    }

    static void tampilInfo(String nama, String nim, String email){
        System.out.println("\nHalo, " + nama + " (" + nim + ")");
        System.out.print("Informasi telah dikirim ke email ");
        System.out.println(email + "\n");
    }

    public static void main(String[] args) {
        String nama = bacaTeks("Nama : ");
        String nim = bacaTeks("NIM : ");
        String email = bacaTeks("Email : ");

        tampilInfo(nama, nim, email);
    }
}
```