

**PRAKTIKUM PARADIGMA PEMROGRAMAN**  
**MODUL 3**



**Disusun oleh :**

**Nama : Fidelia Ping**

**NIM : 245410012**

**Kelas : Informatika 1**

**PROGRAM STUDI INFORMATIKA**  
**PROGRAM SARJANA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA**  
**YOGYAKARTA**  
**2025**

## MODUL 3

### KELAS DAN OBJEK

#### A. TUJUAN PRAKTIKUM

1. Mengorganisir kode: Method membantu memecah kode menjadi bagian-bagian yang lebih kecil dan terorganisir, sehingga lebih mudah dibaca dan dipahami.
2. Mengurangi duplikasi kode: Dengan menggunakan method, Anda dapat menghindari duplikasi kode yang sama di beberapa tempat dalam program.
3. Meningkatkan reusabilitas: Method dapat digunakan kembali di beberapa tempat dalam program, sehingga menghemat waktu dan usaha.
4. Meningkatkan modularitas: Method membantu memisahkan kode menjadi modul-modul yang independen, sehingga lebih mudah untuk memodifikasi dan memelihara kode.
5. Meningkatkan keterbacaan: Method membuat kode lebih mudah dibaca dan dipahami, karena setiap method memiliki nama yang deskriptif dan fungsi yang jelas.
6. Meningkatkan efisiensi: Method dapat membantu meningkatkan efisiensi program dengan mengurangi jumlah kode yang perlu ditulis dan dipelihara. Tuliskan tujuan praktikum pada pertemuan ini

#### B. PEMBAHASAN LISTING

##### Praktik

```
public class ContohMethod {  
    // Method tanpa parameter dan tanpa return  
    public void cetakPesan() {  
        System.out.println("Halo, ini contoh method tanpa parameter  
dan tanpa return!");  
    }  
  
    // Method dengan parameter  
    public void cetakNama(String nama) {  
        System.out.println("Halo, " + nama + "!");  
    }  
  
    // Method dengan return  
    public String getPesan() {  
        return "Halo, ini contoh method dengan return!";  
    }  
  
    // Method dengan parameter dan return  
    public int tambah(int a, int b) {  
        return a + b;  
    }  
  
    public static void main(String[] args) {  
        ContohMethod contoh = new ContohMethod();  
        contoh.cetakPesan(); // Memanggil method tanpa parameter dan  
tanpa return  
    }  
}
```

```

        contoh.cetakNama("UTDI"); // Memanggil method dengan
parameter
        System.out.println(contoh.getPesan()); // Memanggil method
dengan return
        int hasil = contoh.tambah(5, 3); // Memanggil method dengan
parameter dan return
        System.out.println("Hasil tambah: " + hasil);
    }
}

```

**Pembahasan :** pada program “ContohMethod” merupakan contoh penerapan berbagai jenis method dalam bahasa pemrograman Java yang digunakan untuk memperlihatkan perbedaan antara method dengan parameter, tanpa parameter, dengan return, dan tanpa return. Program ini terdiri dari empat method utama, masing-masing dengan karakteristik dan fungsi berbeda. Pertama, terdapat method cetakPesan() yang merupakan method tanpa parameter dan tanpa return. Method ini tidak menerima data dari luar dan juga tidak mengembalikan nilai apa pun, melainkan hanya menjalankan satu perintah yaitu menampilkan pesan “Halo, ini contoh method tanpa parameter dan tanpa return!” ke layar. Method jenis ini biasanya digunakan untuk menampilkan informasi atau melakukan suatu tindakan sederhana tanpa membutuhkan input atau output. Selanjutnya, method kedua yaitu cetakNama(String nama) merupakan contoh method dengan parameter tetapi tanpa return. Method ini membutuhkan satu input berupa parameter bertipe String bernama nama, dan ketika dijalankan, method ini akan mencetak sapaan yang disesuaikan dengan nama yang dikirim, misalnya jika parameter yang dikirim adalah “UTDI”, maka hasil yang tampil di layar adalah “Halo, UTDI!”. Method seperti ini berguna untuk menerima data dari pengguna atau dari bagian lain program, lalu memprosesnya tanpa harus mengembalikan nilai. Method ketiga dalam program ini yaitu getPesan() adalah contoh method dengan return tetapi tanpa parameter. Berbeda dengan dua method sebelumnya, method ini tidak memerlukan input apa pun, namun akan mengembalikan sebuah nilai berupa teks (String) menggunakan pernyataan return. Nilai yang dikembalikan kemudian ditampilkan di layar melalui perintah System.out.println(contoh.getPesan()); sehingga output yang muncul adalah “Halo, ini contoh method dengan return!”. Konsep return ini penting dalam pemrograman karena memungkinkan suatu method memberikan hasil perhitungan atau data kepada bagian lain dari program. Terakhir, method keempat yaitu tambah(int a, int b) merupakan contoh method dengan parameter dan juga return. Method ini menerima dua input bertipe bilangan bulat (integer), menjumlahkan keduanya, lalu mengembalikan hasil penjumlahannya dengan pernyataan return a + b;. Ketika method ini dipanggil melalui int hasil = contoh.tambah(5, 3);, hasil penjumlahan 5 dan 3 akan dikembalikan ke variabel hasil, yang kemudian ditampilkan di layar menjadi “Hasil tambah: 8”. Keempat method ini dipanggil secara berurutan di dalam method main(), dimulai dari pemanggilan cetakPesan(), cetakNama("UTDI"), getPesan(), dan tambah(5, 3).

```

Halo, ini contoh method tanpa parameter dan tanpa return!
Halo, UTDI!
Halo, ini contoh method dengan return!
Hasil tambah: 8
Press any key to continue . . .

```

**Pembahasan** : dari hasil program tersebut dapat disimpulkan bahwa program ini secara jelas menunjukkan empat bentuk dasar method dalam Java, yaitu method tanpa parameter dan tanpa return yang hanya menjalankan perintah, method dengan parameter tanpa return yang menerima masukan namun tidak mengembalikan nilai, method dengan return tanpa parameter yang memberikan hasil tanpa memerlukan input, serta method dengan parameter dan return yang menerima data sekaligus menghasilkan keluaran. Dengan memahami contoh ini, dapat dipelajari bahwa penggunaan method memberikan struktur yang lebih baik pada program, memungkinkan pengelolaan kode yang lebih mudah, serta membantu programmer untuk membuat program yang efisien, modular, dan mudah dikembangkan.

## Latihan

```
import java.util.Scanner;

public class ContohMethod1 {
    // Method tanpa parameter dan tanpa return
    public void cetakPesan() {
        System.out.println("Halo, ini contoh method tanpa parameter
dan tanpa return!");
    }

    // Method dengan parameter
    public void cetakNama(String nama) {
        System.out.println("Halo, " + nama + "!");
    }

    // Method dengan return
    public String getPesan() {
        return "Halo, ini contoh method dengan return!";
    }

    // Method dengan parameter dan return
    public int tambah(int a, int b) {
        return a + b;
    }

    public static void main(String[] args) {
        ContohMethod contoh = new ContohMethod();
        Scanner scanner = new Scanner(System.in);

        System.out.println("Pilih menu:");
        System.out.println("1. Cetak pesan");
        System.out.println("2. Cetak nama");
        System.out.println("3. Get pesan");
        System.out.println("4. Tambah dua bilangan");
        System.out.print("Masukkan pilihan: ");
        int pilihan = scanner.nextInt();
        scanner.nextLine(); // Consume newline left-over

        switch (pilihan) {
            case 1:
                contoh.cetakPesan();
                break;
            case 2:
                System.out.print("Masukkan nama: ");
                String nama = scanner.nextLine();
```

```

        contoh.cetakNama(nama);
        break;
    case 3:
        System.out.println(contoh.getPesan());
        break;
    case 4:
        System.out.print("Masukkan bilangan pertama: ");
        int a = scanner.nextInt();
        System.out.print("Masukkan bilangan kedua: ");
        int b = scanner.nextInt();
        int hasil = contoh.tambah(a, b);
        System.out.println("Hasil tambah: " + hasil);
        break;
    default:
        System.out.println("Pilihan tidak valid");
    }

    scanner.close();
}
}

```

**Pembahasan :** Program di atas berjudul “ContohMethod1”, merupakan contoh penerapan method dinamis dengan parameter, tanpa parameter, dengan return, dan tanpa return menggunakan bahasa Java. Program ini juga memperlihatkan bagaimana penggunaan input dari pengguna (user input) melalui kelas Scanner, serta implementasi struktur kontrol switch-case untuk memilih menu secara interaktif. Program ini merupakan versi pengembangan dari contoh method statis, di mana seluruh data sebelumnya ditentukan langsung di dalam kode. Kini, dengan penggunaan input dinamis, pengguna dapat memilih menu dan memberikan data sesuai keinginannya saat program dijalankan.

```

Pilih menu:
1. Cetak pesan
2. Cetak nama
3. Get pesan
4. Tambah dua bilangan
Masukkan pilihan: 1
Halo, ini contoh method tanpa parameter dan tanpa return!
Press any key to continue . . .

```

```

Pilih menu:
1. Cetak pesan
2. Cetak nama
3. Get pesan
4. Tambah dua bilangan
Masukkan pilihan: 2
Masukkan nama: UTDI
Halo, UTDI!
Press any key to continue . . .

```

```

Pilih menu:
1. Cetak pesan
2. Cetak nama
3. Get pesan
4. Tambah dua bilangan
Masukkan pilihan: 3
Halo, ini contoh method dengan return!
Press any key to continue . . . |

```

```
Pilih menu:
1. Cetak pesan
2. Cetak nama
3. Get pesan
4. Tambah dua bilangan
Masukkan pilihan: 4
Masukkan bilangan pertama: 5
Masukkan bilangan kedua: 5
Hasil tambah: 10
Press any key to continue . . . |
```

**Pembahasan :** Pengguna diminta memasukkan pilihan berupa angka 1–4. Setelah pengguna mengetik angka dan menekan Enter, nilai tersebut dibaca oleh program dan disimpan dalam variabel pilihan. Program kemudian menggunakan struktur switch(pilihan) untuk menentukan perintah apa yang akan dijalankan.

**Jika pengguna memilih 1,** program memanggil method cetakPesan(), yang menampilkan pesan standar “Halo, ini contoh method tanpa parameter dan tanpa return!”.

**Jika pengguna memilih 3,** program memanggil method getPesan() dan menampilkan nilai yang dikembalikannya, yaitu “Halo, ini contoh method dengan return!”.

**Jika pengguna memilih 4,** program meminta dua angka dari pengguna. Setelah kedua angka dimasukkan, program memanggil method tambah(a, b) dan menampilkan hasil penjumlahannya. Misalnya, jika pengguna memasukkan 5 dan 3, maka hasilnya akan ditampilkan sebagai “Hasil tambah: 8”.

**Jika pengguna memasukkan angka selain 1–4,** maka program menampilkan pesan “Pilihan tidak valid”.

## C. PEMBAHASAN TUGAS

### Tugas

```
import java.util.Scanner;
public class Kalkulator_Menu {
    public int tambah(int a, int b) {
        return a + b;
    }
    public int kurang(int a, int b) {
        return a - b;
    }
    public int kali(int a, int b) {
        return a * b;
    }
    public double bagi(double a, double b) {
        if (b != 0) {
            return a / b;
        } else {
            System.out.println("Error: Pembagian dengan nol!");
            return 0;
        }
    }
}
public static void main(String[] args) {
    Kalkulator_Menu kalkulator = new Kalkulator_Menu();
    Scanner scanner = new Scanner(System.in);
    while (true) {
        System.out.println("\nMenu Kalkulator:");
        System.out.println("1. Tambah");
        System.out.println("2. Kurang");
        System.out.println("3. Kali");
        System.out.println("4. Bagi");
```

```

        System.out.println("5. Keluar");
        System.out.print("Pilih menu: ");
        int pilihan = scanner.nextInt();
        if (pilihan == 5) {
            System.out.println("Terima kasih!");
            break;
        }
        System.out.print("Masukkan bilangan pertama: ");
        double a = scanner.nextDouble();
        System.out.print("Masukkan bilangan kedua: ");
        double b = scanner.nextDouble();
        switch (pilihan) {
            case 1:
                System.out.println("Hasil      tambah:      "      +
kalkulator.tambah((int) a, (int) b));
                break;
            case 2:
                System.out.println("Hasil      kurang:      "      +
kalkulator.kurang((int) a, (int) b));
                break;
            case 3:
                System.out.println("Hasil      kali:      "      +
kalkulator.kali((int) a, (int) b));
                break;
            case 4:
                System.out.println("Hasil      bagi:      "      +
kalkulator.bagi(a, b));
                break;
            default:
                System.out.println("Pilihan tidak valid");
        }
    }
    scanner.close();
}
}

```

**Pembahasan :** Program berjudul **“Kalkulator\_Menu”** ini merupakan sebuah contoh penerapan konsep **method (fungsi)** dan **penggunaan struktur kontrol menu interaktif** dalam bahasa pemrograman Java. Program ini dirancang untuk menjalankan operasi aritmetika dasar — yaitu **penjumlahan, pengurangan, perkalian, dan pembagian** — dengan melibatkan masukan dari pengguna secara langsung melalui kelas Scanner. Program ini juga menggunakan **perulangan (looping)** dan **percabangan (switch-case)** untuk memproses pilihan menu yang dimasukkan pengguna, sehingga pengguna dapat melakukan perhitungan berulang kali tanpa harus menjalankan ulang program dari awal. Struktur utama program terdiri dari dua bagian besar, yaitu **deklarasi method operasi matematika** dan **blok utama (method main)** yang berfungsi sebagai titik awal eksekusi program. Pada bagian awal, terdapat empat buah method yang masing-masing mewakili operasi aritmetika dasar. Pertama adalah method `tambah(int a, int b)` yang berfungsi untuk melakukan penjumlahan dua bilangan integer. Method ini menerima dua parameter bertipe `int`, kemudian mengembalikan hasilnya dengan perintah `return a + b;`. Kedua, terdapat method `kurang(int a, int b)` yang memiliki struktur serupa, namun digunakan untuk menghitung selisih antara dua bilangan. Hasil pengurangan diperoleh melalui ekspresi `a - b`. Ketiga, method `kali(int a, int b)` digunakan untuk mengalikan dua bilangan integer dan mengembalikan hasilnya melalui `return a * b;`. Sementara itu, operasi pembagian ditangani oleh method `bagi(double a, double b)` yang menggunakan tipe data

**double** agar hasilnya dapat berupa bilangan desimal. Di dalam method ini terdapat **kondisi pengaman (if statement)** untuk mencegah kesalahan pembagian dengan nol. Jika pengguna memasukkan nilai b sama dengan nol, program akan menampilkan pesan "Error: Pembagian dengan nol!" dan mengembalikan nilai 0 sebagai tanda kesalahan. Namun, jika nilai b tidak sama dengan nol, program akan menghitung hasil bagi a / b dan mengembalikannya. Logika ini menunjukkan penerapan konsep **pengecekan kondisi dan pengendalian kesalahan sederhana** di dalam method. Bagian selanjutnya adalah method main() yang menjadi bagian utama dari program. Pada bagian ini, pertama-tama dibuat **objek dari kelas Kalkulator\_Menu** dengan perintah `Kalkulator_Menu kalkulator = new Kalkulator_Menu();` yang berfungsi untuk memanggil method-method yang telah dibuat sebelumnya. Kemudian dibuat objek Scanner `scanner = new Scanner(System.in);` untuk menerima input dari pengguna melalui keyboard. Setelah itu, program masuk ke dalam sebuah **looping while(true)**, artinya program akan terus berjalan dan menampilkan menu sampai pengguna memilih opsi untuk keluar.

```
Menu Kalkulator:
1. Tambah
2. Kurang
3. Kali
4. Bagi
5. Keluar
Pilih menu: 1
Masukkan bilangan pertama: 3
Masukkan bilangan kedua: 3
Hasil tambah: 6

Menu Kalkulator:
1. Tambah
2. Kurang
3. Kali
4. Bagi
5. Keluar
Pilih menu: 2
Masukkan bilangan pertama: 4
Masukkan bilangan kedua: 2
Hasil kurang: 2

Menu Kalkulator:
1. Tambah
2. Kurang
3. Kali
4. Bagi
5. Keluar
Pilih menu: 3
```

```

Masukkan bilangan pertama: 5
Masukkan bilangan kedua: 5
Hasil kali: 25

Menu Kalkulator:
1. Tambah
2. Kurang
3. Kali
4. Bagi
5. Keluar
Pilih menu: 4
Masukkan bilangan pertama: 100
Masukkan bilangan kedua: 4
Hasil bagi: 25.0

Menu Kalkulator:
1. Tambah
2. Kurang
3. Kali
4. Bagi
5. Keluar
Pilih menu: 5
Terima kasih!
Press any key to continue . . .

```

**Pembahasan :** Kemudian program menunggu input pengguna berupa angka yang mewakili pilihan operasi. Jika pengguna memilih angka 5, program menampilkan pesan “Terima kasih!” dan keluar dari perulangan dengan perintah `break`; Namun, jika pengguna memilih angka 1 sampai 4, program akan meminta dua buah bilangan untuk dioperasikan. Input ini disimpan dalam variabel `a` dan `b`.

Jika pengguna memilih **1**, program akan menjalankan `kalkulator.tambah((int)a, (int)b)` untuk menghitung penjumlahan dua bilangan dan menampilkan hasilnya.

Jika memilih **2**, program menjalankan `kalkulator.kurang((int)a, (int)b)` untuk menghitung selisih kedua bilangan.

Jika memilih **3**, program memanggil `kalkulator.kali((int)a, (int)b)` untuk menghitung hasil perkalian.

Jika memilih **4**, program memanggil `kalkulator.bagi(a, b)` untuk menghitung hasil pembagian dua bilangan desimal.

Apabila pengguna memasukkan angka selain 1–5, program menampilkan pesan “Pilihan tidak valid”.

## Tugas 2

```

import java.util.Scanner;

public class Konverter {

    Scanner input = new Scanner(System.in);

    //tanpa parameter & tanpa return
    public void konversiSuhu1() {
        System.out.print("Masukkan suhu dalam Celsius: ");
        double celsius = input.nextDouble();
        double fahrenheit = (celsius * 9/5) + 32;
        System.out.println("Hasil konversi: " + fahrenheit + "°F");
    }

    //dengan parameter & tanpa return
    public void konversiSuhu2(double celsius) {
        double fahrenheit = (celsius * 9/5) + 32;
    }
}

```

```

        System.out.println("Hasil konversi: " + fahrenheit + "°F");
    }

    //tanpa parameter & dengan return
    public double konversiPanjang1() {
        System.out.print("Masukkan panjang dalam meter: ");
        double meter = input.nextDouble();
        double kilometer = meter / 1000;
        return kilometer;
    }

    //dengan parameter & dengan return
    public double konversiPanjang2(double meter) {
        double kilometer = meter / 1000;
        return kilometer;
    }

    public static void main(String[] args) {
        Konverter konverter = new Konverter();

        System.out.println("=== KONVERSI SUHU ===");
        konverter.konversiSuhu1();

        System.out.println("\n=== KONVERSI SUHU ===");
        konverter.konversiSuhu2(30);

        System.out.println("\n=== KONVERSI PANJANG ===");
        double hasil1 = konverter.konversiPanjang1();
        System.out.println("Hasil konversi: " + hasil1 + " kilometer");

        System.out.println("\n=== KONVERSI PANJANG ===");
        double hasil2 = konverter.konversiPanjang2(2500); // contoh input
2500 meter
        System.out.println("Hasil konversi: " + hasil2 + " kilometer");
    }
}

```

**Pembahasan :** Program Konverter merupakan program yang dibuat untuk melakukan dua jenis konversi, yaitu konversi suhu dari Celsius ke Fahrenheit dan konversi panjang dari meter ke kilometer. program ini juga dirancang untuk menunjukkan **empat macam bentuk metode**.

**Metode Tanpa Parameter dan Tanpa Return,** Pada metode ini, program meminta pengguna untuk **memasukkan nilai suhu dalam satuan Celsius** melalui keyboard menggunakan Scanner. Hasil konversi ditampilkan langsung di layar dengan perintah `System.out.println()`. Metode ini tidak memiliki **parameter**

**Metode Dengan Parameter dan Tanpa Return,** Metode ini dinamakan **konversiSuhu2(double celsius)**. metode ini **menerima nilai Celsius melalui parameter** yang dikirim saat pemanggilan metode. Nilai tersebut kemudian dikonversi ke Fahrenheit dengan rumus yang sama, dan hasilnya langsung ditampilkan.

**Metode Tanpa Parameter dan Dengan Return,** Metode ini dinamakan **konversiPanjang1()**. pengguna diminta untuk **memasukkan nilai panjang dalam meter**, kemudian dihitung untuk diubah menjadi kilometer. Setelah

dihitung, hasil konversi **tidak langsung ditampilkan**, melainkan **dikembalikan (return)** kepada pemanggil metode dalam bentuk double.

**Metode Dengan Parameter dan Dengan Return**, Metode ini dinamakan **konversiPanjang2(double meter)**. Metode ini menerima **nilai meter sebagai parameter**, kemudian menghitung hasil konversinya ke kilometer menggunakan rumus yang sama, dan **mengembalikan hasilnya (return)** ke pemanggil metode.

```
=== KONVERSI SUHU ===  
Masukkan suhu dalam Celsius: 30  
Hasil konversi: 86.0°F  
  
=== KONVERSI SUHU ===  
Hasil konversi: 86.0°F  
  
=== KONVERSI PANJANG ===  
Masukkan panjang dalam meter: 5  
Hasil konversi: 0.005 kilometer  
  
=== KONVERSI PANJANG ===  
Hasil konversi: 2.5 kilometer  
Press any key to continue . . . |
```

**Pembahasan :** Setelah program dijalankan dengan input suhu 30°C dan panjang 5 meter, program menampilkan hasil yang menunjukkan cara kerja dari empat jenis metode berbeda dalam Java. Pada bagian konversi suhu tanpa parameter dan tanpa return, pengguna diminta memasukkan nilai suhu dalam Celsius, yaitu 30, kemudian program menghitung hasil konversi dengan rumus  $(Celsius \times 9/5) + 32$ . Dari perhitungan tersebut diperoleh hasil 86.0°F, yang langsung ditampilkan di layar. Selanjutnya, pada metode dengan parameter dan tanpa return, nilai suhu 30°C dikirim melalui parameter tanpa input tambahan dari pengguna, dan menghasilkan keluaran yang sama yaitu 86.0°F, karena menggunakan rumus konversi yang identik. Setelah itu, program melanjutkan ke konversi panjang tanpa parameter dan dengan return, di mana pengguna memasukkan nilai panjang 5 meter. Program kemudian menghitung hasil konversi menggunakan rumus  $meter / 1000$ , menghasilkan nilai 0.005 kilometer, yang dikembalikan ke fungsi utama (main) untuk ditampilkan di layar. Terakhir, pada metode dengan parameter dan dengan return, nilai panjang 5 meter juga dikirim langsung melalui parameter, dihitung dengan rumus yang sama, dan menghasilkan output 0.005 kilometer.

#### D. KESIMPULAN

Saya dapat mengetahui dari 4 metode, dan 4 metode tersebut dapat dimasukan kedalam 1 program. Dan keempat nya memiliki fungsi dan cara kerja yang berbeda dalam menerima data serta menampilkan hasil.