



UNIVERSITAS
ISLAM
INDONESIA

FUNDAMEN PENGEMBANGAN APLIKASI

Pertemuan 5

Void Method & Non Void Method

TOPIK MATERI



UNIVERSITAS
ISLAM
INDONESIA

1. Method
2. Struktur Method
3. Studi Kasus Void Method
4. Studi Kasus Non Void Method



UNIVERSITAS
ISLAM
INDONESIA

Method

Method

Definisi



UNIVERSITAS
ISLAM
INDONESIA

- Method adalah **blok program** yang akan **dijalankan saat dipanggil**. Method **dapat menerima data** dan melakukan beberapa aksi. Method juga dikenal sebagai fungsi/**function** (w3school.com)
 - **Tujuan**: Untuk **menggunakan kode secara berulang** (mendefinisikan kode sekali dan menggunakannya berkali-kali)
- Method adalah **kumpulan statement** yang melakukan suatu tugas tertentu. Method juga **dapat mengembalikan suatu nilai** ke pemanggilnya (www.geeksforgeeks.org)
 - **Tujuan**: Untuk **menghemat waktu** (dapat menggunakan suatu perintah tanpa harus menuliskan lagi)

Method

Analogi



UNIVERSITAS
ISLAM
INDONESIA

Buat Kopi

Siapkan Bubuk Kopi

Siapkan Cangkir

Siapkan Air Panas

Siapkan Sendok

Masukkan Kopi ke Cangkir

Tuangkan Air Panas ke Cangkir

Aduk Kopi

- Perhatikan alur proses di samping! Kira-kira apa yang ingin saya buat?
- Mana yang lebih enak disebutkan jika Anda ingin seseorang membuatkan kopi untuk Anda?
 - Proses-proses dalam membuat kopi atau cukup menyebutkan "**buatkan kopi**"?
- Pada contoh ini, buat kopi adalah suatu **method**, di mana terdapat proses-proses yang dapat dianalogikan sebagai **statement**

Method, Fungsi, dan Prosedur

Hubungan dan Komparasi



UNIVERSITAS
ISLAM
INDONESIA

- Pada pemrograman Java, **method** adalah blok program terpisah (di luar blok program utama) yang dapat digunakan untuk menyelesaikan masalah khusus
- **Fungsi** adalah method yang **mengembalikan suatu nilai** (juga disebut sebagai **Non Void Method**)
- Beberapa sumber juga menyebut Non Void Method dengan istilah **Return Method**
- **Prosedur** adalah method yang **tidak mengembalikan suatu nilai** (juga disebut sebagai **Void Method**)



UNIVERSITAS
ISLAM
INDONESIA

Struktur Method

Struktur Method

Void

```
static void <namaMethod> (<parameter>){  
    <Statement>  
}
```

- void** : keyword yang menandakan bahwa method yang dibuat merupakan method void tanpa nilai kembalian
- NamaMethod** : Nama method (spesifik, jelas, dan sesuai dengan aturan penamaan)
- Parameter** : Parameter yang digunakan pada method (opsional)

Struktur Method

Void (contoh)

```
static void <namaMethod> (<parameter>){  
    <Statement>  
}
```

Contoh:

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

```
public static void main(String[] args) {  
    sapa();  
}
```

Struktur Method

Non-Void

```
static <tipoData> <namaMethod> (<parameter>){  
    <Statement>  
    return <nilaiBalik>;  
}
```

TipoData : Tipe data yang dikembalikan, jika tidak ada nilai yang dikembalikan (Non Void Method) maka diisi dengan kata "void"

NamaMethod : Nama method (spesifik, jelas, dan sesuai dengan aturan penamaan)

Parameter : Parameter yang digunakan pada method (opsional)

NilaiBalik : Nilai balik sesuai tipe data yang didefinisikan (hanya digunakan untuk **Non Void Method**)

Struktur Method

Non-Void



UNIVERSITAS
ISLAM
INDONESIA

```
static int <namaMethod> (<parameter>){  
    <Statement>  
    return <nilaiBalik>;  
}
```

TipeData

- Bisa diisi tipe data primitif sesuai dengan nilai balik (boolean, byte, short, int, long, float, double, atau char)
- Bisa diisi tipe data reference/bentukan sesuai dengan nilai balik (String, ArrayList, dll)
- Bisa diisi dengan kata "void" jika method tidak memiliki nilai balik (procedure)

Struktur Method

Non-Void



UNIVERSITAS
ISLAM
INDONESIA

```
static int luas (<parameter>){  
    <Statement>  
    return <nilaiBalik>;  
}
```

NamaMethod

- Dapat menggambarkan/mendeskripsikan method
- Bisa terdiri dari huruf, angka, dan garis bawah (_)
- Karakter pertama tidak boleh angka/garis bawah
- Tidak boleh mengandung spasi
- Bisa menggunakan teknik CamelCase untuk menggantikan spasi

Struktur Method

Non-Void

```
static int luas (int panjang, int lebar){  
    <Statement>  
    return <nilaiBalik>;  
}
```

Parameter

- Parameter adalah informasi yang dikirimkan pada method dan berperan sebagai variabel pada method (parameter formal pada definisi fungsi/prosedur)
- Format penulisan parameter: **<tipeData> <namaParameter>**
- Tipe data dapat terdiri dari tipe data primitif, reference, atau tipe data bentukan.
- Parameter bersifat opsional (bisa tidak ada, bisa ada 1, bisa lebih dari 1)
- Untuk parameter lebih dari 1, harus dipisahkan dengan koma

Struktur Method

Non-Void (contoh)

```
static int luas (int panjang, int lebar){  
    int luas = panjang * lebar;  
    return luas;  
}
```

NilaiBalik

- Nilai balik adalah nilai akhir yang dikirimkan oleh sebuah method (layaknya sebuah fungsi)
- Penulisan nilai balik harus diawali dengan menuliskan **return**
- Tiap Non Void Method hanya dapat memberikan satu nilai balik
- Untuk Void Method, nilai balik tidak diperlukan

Review Parameter

Pass by Value dan Pass by Reference

- **Pass by Value**

Memberikan salinan nilai variabel ke sebuah method

- **Pass by Reference**

Memberikan alamat memory stack sebuah variabel ke sebuah method

Review Parameter

Pass by Value dan Pass by Reference

■ Pass by Value

Memberikan salinan nilai variabel ke sebuah method

■ Pass by Reference

Memberikan alamat memory stack sebuah variabel ke sebuah method

```
int n = 3
```

```
static void kuadrat(int i){  
    i = i * i;  
    System.out.println(i);  
}
```

- Program akan menampilkan nilai **9**
- Nilai variabel **n** akan tetap **3**

Review Parameter

Pass by Value dan Pass by Reference

■ Pass by Value

Memberikan salinan nilai variabel ke sebuah method

■ Pass by Reference

Memberikan alamat memory stack sebuah variabel ke sebuah method

`int n = 3`

```
static void kuadrat(int i){  
    i = i * i;  
    System.out.println(i);  
}
```

- Program akan menampilkan nilai **9**
- Nilai variabel **n** akan mengalami perubahan dan akan bernilai **9**

Review Parameter

Pass by Value dan Pass by Reference

- **Pass by Value**

Memberikan salinan nilai variabel ke sebuah method

- **Pass by Reference**

Memberikan alamat memory stack sebuah variabel ke sebuah method

- Secara alami pemrograman **Java** menggunakan teknik **Pass by Value**

Pemanggilan Method

Void Method

- Dituliskan nama methodnya dan diikuti dengan **argumen** (nilai parameter) jika method tersebut menggunakan parameter

```
static void menyapa(String nama){  
    System.out.println("Halo " + nama);  
}
```

```
public static void main(String[] args) {  
    menyapa("Mahasiswa");  
}
```

Pemanggilan Method

Non Void Method

- Dituliskan nama methodnya dan diikuti dengan **argumen** (nilai parameter) jika method tersebut menggunakan parameter
- Karena menggunakan nilai balik (yang harus ditampung atau dieksekusi), maka non void method tidak dapat berdiri sendiri

```
static int kuadrat(int n){  
    return n*n;  
}
```

```
public static void main(String[] args) {  
    System.out.println(kuadrat(5));  
}
```


Contoh 1

- Membuat method untuk menampilkan teks dengan format sebagai berikut:

Halo <panggilan> <nama>

- Terdapat 2 parameter
 - Parameter pertama mendefinisikan panggilan dengan menggunakan kode dalam bentuk bilangan bulat dengan aturan sebagai berikut
1 : Mas; 2 : Mbak; 3 : Pak; 4 : Bu
 - Parameter kedua mendefinisikan nama
- Akan muncul seperti gambar di samping jika dimasukkan argument berikut:

Parameter	panggilan	nama
Argumen	1	Andi
	3	Budi
	2	Cika

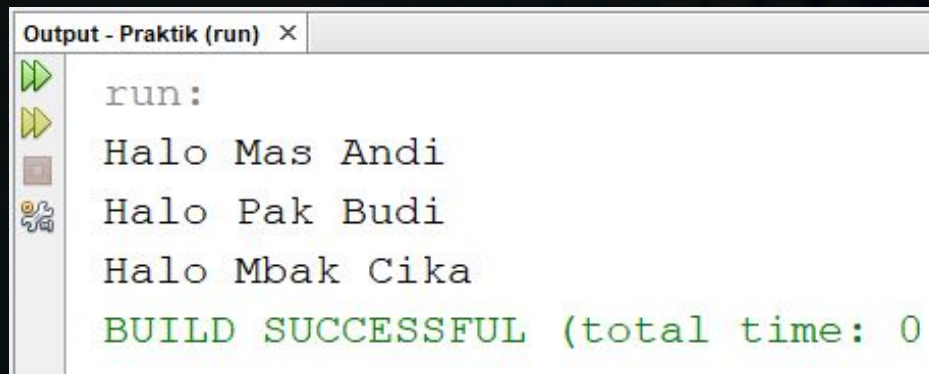
```
Output - Praktik (run) x
run:
Halo Mas Andi
Halo Pak Budi
Halo Mbak Cika
BUILD SUCCESSFUL (total time: 0
```

Contoh 1

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

Contoh 1

```
public static void main(String[] args) {  
    menyapa(1, "Andi");  
    menyapa(3, "Budi");  
    menyapa(2, "Cika");  
}
```

A screenshot of an IDE's output window titled "Output - Praktik (run) x". The window shows the execution output of the Java code. On the left side of the window, there are three green right-pointing arrows and a red square icon. The output text is as follows:
run:
Halo Mas Andi
Halo Pak Budi
Halo Mbak Cika
BUILD SUCCESSFUL (total time: 0
The text "BUILD SUCCESSFUL" is highlighted in green.

Contoh 2

- Membuat method untuk menggambar karakter pagar (#) dan spasi
- Terdapat 3 parameter
 - Parameter pertama mendefinisikan berapa banyak karakter pagar akan ditulis
 - Parameter kedua mendefinisikan berapa banyak karakter spasi akan ditulis setelah karakter pagar
 - Parameter ketiga mendefinisikan berapa banyak karakter pagar akan ditulis setelah karakter spasi
- Misal saya memberikan argumen (3, 2, 3), maka akan menampilkan:
###
- Misal saya memberikan argumen (3, 0, 3), maka akan menampilkan:
#####

Contoh 2

Misal saya memasukkan kombinasi dari argumen-argumen berikut:

Parameter	?	??	??
Argumen	6	0	0
	2	4	1
	6	0	0
	2	3	2
	2	3	2
	0	0	6

maka akan muncul seperti gambar di samping

```

Output - Praktik (run) x
run:
#####
##      #
#####
##      ##
##      ##
#####
BUILD SUCCESSFUL (total time: 0
  
```

Contoh 2

```
static void gambar(int a, int b, int c){  
    for(int i = 0; i < a; i++){  
        System.out.print('#');  
    }  
    for(int i = 0; i < b; i++){  
        System.out.print(' ');  
    }  
    for(int i = 0; i < c; i++){  
        System.out.print('#');  
    }  
    System.out.println();  
}
```

Contoh 2

```
public static void main(String[] args) {  
    gambar(6, 0, 0);  
    gambar(2, 4, 1);  
    gambar(6, 0, 0);  
    gambar(2, 3, 2);  
    gambar(2, 3, 2);  
    gambar(0, 0, 6);  
}
```

Contoh 3

- Membuat method yang menghasilkan nilai balik berupa total penjumlahan dari 3 bilangan bulat
- Terdapat 3 parameter
 - Parameter pertama mendefinisikan bilangan pertama
 - Parameter kedua mendefinisikan bilangan kedua
 - Parameter ketiga mendefinisikan bilangan ketiga
- Nilai balik kemudian kemudian disimpan dalam variabel bernama total
- Variabel total kemudian ditampilkan sebagai keluaran
- Jika saya memasukkan argumen dengan nilai (5, 2, 3), keluaran yang muncul adalah 10
- Jika saya memasukkan argumen dengan nilai (-7, -8, -9), keluaran yang muncul adalah -24

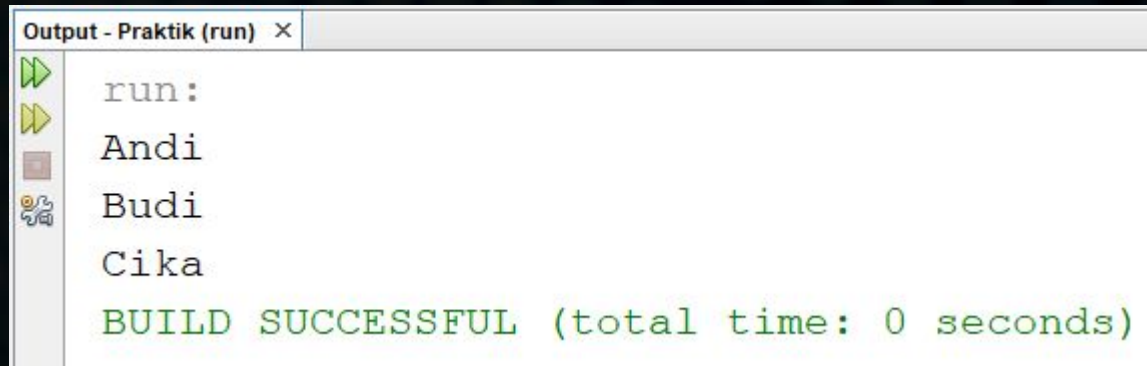
Contoh 3

```
static int penjumlahan(int a, int b, int c){  
    int total = a + b + c;  
    return total;  
}
```

```
public static void main(String[] args) {  
    int total = penjumlahan(5, 2, 3);  
    System.out.println(total);  
  
    total = penjumlahan(-7, -8, -9);  
    System.out.println(total);  
}
```

Contoh 4

- Membuat method tanpa nilai balik yang dapat menampilkan nilai elemen suatu larik secara beruntun
- Hanya menggunakan 1 parameter yang mendefinisikan larik yang akan ditampilkan
- Gunakan perintah **<namaLarik>.length** untuk mengetahui panjang larik
- Jika saya memiliki larik yang berisi elemen "Andi", "Budi", dan "Cika", keluaran yang dihasilkan adalah sebagai berikut



```
Output - Praktik (run) x
run:
Andi
Budi
Cika
BUILD SUCCESSFUL (total time: 0 seconds)
```

Contoh 4

```
static void tampilLarik(String larik[]){  
    for(int i=0; i < larik.length; i++){  
        System.out.println(larik[i]);  
    }  
}  
  
public static void main(String[] args) {  
    String nama[] = {"Andi", "Budi", "Cika"};  
    tampilLarik(nama);  
}
```



UNIVERSITAS
ISLAM
INDONESIA

Studi Kasus Void Method

Praktik 1

Nilai Tempat



UNIVERSITAS
ISLAM
INDONESIA

- Buatlah method yang dapat menampilkan nilai tempat (satuan, puluhan, ratusan, ribuan) dari argumen yang diberikan
- Hanya membutuhkan satu parameter yang mendefinisikan nilai bilangan bulat lebih dari nol
- Jika diberikan argumen dalam rentang 1 – 9.999, akan dihasilkan nilai balik sesuai dengan nilai tempatnya, selain itu akan dihasilkan nilai balik “Salah”.
- Rentang nilai:
 - 1 – 9 : Satuan 100 – 999 : Ratusan
 - 10 – 99 : Puluhan 1.000 – 9.999 : Ribuan

Praktik 1

Nilai Tempat



UNIVERSITAS
ISLAM
INDONESIA

Misal saya memasukkan argumen-argumen berikut:

Parameter	?
Argumen	-5
	7
	99
	123
	9876
	10000

maka akan tampil keluaran seperti pada gambar di samping

```
Output - Praktik (run) x
run:
Salah
Satuan
Puluhan
Ratusan
Ribuan
Salah
BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktik 1

Nilai Tempat



UNIVERSITAS
ISLAM
INDONESIA

```
// lengkapi bagian ...
static void ...(... ...){
    if(n >= 1 && n <= 9)
        System.out.println("Satuan");
    else if(n >= 10 && n <= 99)
        System.out.println("Puluhan");
    else if(n >= 100 && n <= 999)
        System.out.println("Ratusan");
    else if(n >= 1000 && n <= 9999)
        System.out.println("Ribuan");
    else
        System.out.println("Salah");
}
```

Tips:

Untuk menyederhanakan baris perintah, jika percabangan (**if/else-if/else**) hanya berisi satu statement, maka **kurung kurawal dapat dihilangkan**

NB:

1. Ekspresi kondisi sebagai syarat percabangan dapat disederhanakan
2. Penyelesaian masalah ini dapat dilakukan dengan pendekatan matematis sehingga berlaku secara generik berapapun digit n

Praktik 1

Nilai Tempat



UNIVERSITAS
ISLAM
INDONESIA

```
public static void main(String[] args) {  
    nilaiTempat(-5);  
    nilaiTempat(7);  
    nilaiTempat(99);  
    nilaiTempat(123);  
    nilaiTempat(9876);  
    nilaiTempat(10000);  
}
```


Praktik 2

Nama dan Nilai



UNIVERSITAS
ISLAM
INDONESIA

- Buatlah method yang dapat menampilkan kombinasi nama dan nilai secara berurutan dari dua larik yang berbeda (larik nama dan larik nilai)
- Terdapat 2 parameter
 - Parameter pertama mendefinisikan larik yang berisi deretan nama
 - Parameter kedua mendefinisikan larik yang berisi deretan nilai
- Nilai elemen larik
 - Nama : Andi, Budi, Cika, Denis, Endro
 - Nilai : 75, 87, 82, 88, 77
- Keluaran yang dihasilkan adalah sebagai berikut:

```
Output - Praktik (run) x
run:
Andi : 75
Budi : 87
Cika : 82
Denis : 88
Endro : 77
BUILD SUCCESSFUL (total time: 0
```

Praktik 2

Nama dan Nilai

// lengkapi bagian ...

```
static void ...(... ..., ... ...){  
    for(int i=0; i < nama.length; i++){  
        System.out.print(nama[i] + " : ");  
        System.out.println(nilai[i]);  
    }  
}  
  
public static void main(String[] args) {  
    String nama[] = {"Andi", "Budi", "Cika", "Denis", "Endro"};  
    int nilai[] = {75, 87, 82, 88, 77};  
    tampil(nama, nilai);  
}
```

Praktik 3

Diagram



UNIVERSITAS
ISLAM
INDONESIA

- Buatlah method yang dapat menampilkan karakter sebanyak nilai tiap elemen pada sebuah larik
- Terdapat 2 parameter
 - Parameter pertama mendefinisikan karakter apa yang akan ditampilkan
 - Parameter kedua mendefinisikan larik yang berisi deretan nilai
- Nilai elemen larik : 5, 9, 6, 8, 7
- Keluaran yang dihasilkan adalah sebagai berikut:

```
Output - Praktik (run) ×
run:
5 : =====
9 : =====
6 : =====
8 : =====
7 : =====
BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktik 3

Diagram



UNIVERSITAS
ISLAM
INDONESIA

```
// lengkapi bagian ...

static void ...(... ..., ... ...){
    for(int i=0; i < nilai.length; i++){
        System.out.print(nilai[i] + " : ");
        for(int n=0; n < nilai[i]; n++){
            System.out.print(karakter);
        }
        System.out.println();
    }
}

public static void main(String[] args) {
    int nilai[] = {5, 9, 6, 8, 7};
    diagram('-', nilai);
}
```




UNIVERSITAS
ISLAM
INDONESIA

Studi Kasus Non Void Method

Praktik 4

Kode



UNIVERSITAS
ISLAM
INDONESIA

- Buatlah method yang memberikan nilai balik dari hasil pengkodean sebuah bilangan bulat di rentang 100.000 sampai 999.999 (di luar rentang ini, sistem akan memberikan nilai 0)
- Hanya terdapat 1 parameter yang mendefinisikan bilangan bulat yang akan dikodekan (kita sebut sebagai “sumber”)
- Aturan pengkodeannya:
 - Sumber ditambah 24
 - Hasilnya dikali 3
 - Hasilnya dikurangi 87
 - Hasilnya dikali 2
 - Hasilnya ditambah 9

Praktik 4

Kode



UNIVERSITAS
ISLAM
INDONESIA

- Misal saya memasukkan argumen-argumen berikut:

Parameter	?
Argumen	12345
	1234567
	123456

- Lalu saya tampilkan nilai balik yang telah diterima, maka akan menghasilkan
- keluaran sebagai berikut

```
Output - Praktik (run) x
run:
0
0
740715
BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktik 4

Kode



UNIVERSITAS
ISLAM
INDONESIA

```
// lengkapi bagian ...
```

```
static int ...(... ...){  
    if(sumber >= 100000 && sumber <= 999999){  
        int hasil = sumber + 24;  
        hasil *= 3;  
        hasil -= 87;  
        hasil *= 2;  
        hasil += 9;  
        return hasil;  
    }  
    else{  
        return 0;  
    }  
}
```


Praktik 4

Kode



UNIVERSITAS
ISLAM
INDONESIA

```
public static void main(String[] args) {  
    System.out.println(kode(12345));  
    System.out.println(kode(1234567));  
    System.out.println(kode(123456));  
}
```

Praktik 4 Plus

Dekode



UNIVERSITAS
ISLAM
INDONESIA

- Dengan teknik yang sama, buatlah method yang digunakan untuk mengembalikan hasil pengkodean di praktik 4
- Perlu diingat, pada method dekode ini tidak diperlukan batasan rentang nilai
- Misal saya memasukkan argumen (5022165) dan argumen (5076189), lalu saya tampilkan nilai balik yang telah diterima, maka akan menghasilkan keluaran sebagai berikut

```
Output - Praktik (run) x
run :
837031
846035
BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktik 4 Plus

Dekode

// lengkapi bagian ...

```
static int dekode(int sumber){  
    .....  
    int hasil = sumber - 9;  
    hasil /= 2;  
    hasil += 87;  
    hasil /= 3;  
    hasil -= 24;  
    return hasil;  
}  
  
public static void main(String[] args) {  
    System.out.println(dekode(5022165));  
    System.out.println(dekode(5076189));  
}
```



UNIVERSITAS
ISLAM
INDONESIA

Praktik 5

Konversi Suhu



UNIVERSITAS
ISLAM
INDONESIA

- Buatlah method yang memberikan nilai balik konversi suhu dari Celcius ke Reamur, Kelvin, atau Fahrenheit
- Terdapat 2 parameter
 - Parameter pertama mendefinisikan kode tujuan konversi. Kodennya adalah 'R' untuk konversi ke Reamur, 'K' untuk konversi ke Kelvin, atau 'F' untuk konversi ke Fahrenheit
 - Parameter kedua mendefinisikan nilai suhu dalam Celcius
- Rumus:
 - $\text{Reamur} = 0,8 \times \text{Celcius}$
 - $\text{Kelvin} = C + 273$
 - $\text{Fahrenheit} = (1,8 \times \text{Celcius}) + 32$

Praktik 5

Konversi Suhu



UNIVERSITAS
ISLAM
INDONESIA

- Misal saya memasukkan argumen-argumen berikut:

Parameter	satuan	C
Argumen	R	123
	K	72
	F	241

- Lalu saya tampilkan nilai balik yang telah diterima, maka akan menghasilkan keluaran sebagai berikut

```
Output - Praktik (run) x
run :
98.4
345
465.8
BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktik 5

Konversi Suhu

// lengkapi bagian ...

```
static void ...(... ..., ... ...){  
    switch(satuan){  
        case 'R' : System.out.println(0.8 * c); break;  
        case 'K' : System.out.println(c + 273); break;  
        case 'F' : System.out.println((1.8 * c) + 32); break;  
    }  
}
```

```
public static void main(String[] args) {  
    konversi('R', 123);  
    konversi('K', 72);  
    konversi('F', 241);  
}
```

Tips:

Integer jika diproses dengan data bertipe double menggunakan operator tertentu akan menghasilkan nilai double

Praktik 6

Faktorial



UNIVERSITAS
ISLAM
INDONESIA

- Buatlah method yang memberikan nilai balik berupa hasil faktorial dari bilangan bulat yang dimasukkan
- Hanya terdapat 1 parameter yang mendefinisikan bilangan bulat yang akan dihitung nilai faktorialnya
- Contoh faktorial $5 = 5 * 4 * 3 * 2 * 1 = 120$
- Misal saya memasukkan argumen (5), argumen (7), dan argumen (11), lalu saya tampilkan nilai balik yang telah diterima, maka akan menghasilkan keluaran sebagai berikut

```
Output - Praktik (run) x
run:
120
5040
39916800
BUILD SUCCESSFUL (total time:
```

Praktik 6

Faktorial

// lengkapi bagian ...

```
static int ...(.. ...){  
    int hasil = 1;  
    for(int i=n; i > 0; i--){  
        hasil *= i;  
    }  
    return hasil;  
}
```

```
public static void main(String[] args) {  
    System.out.println(ftr(5));  
    System.out.println(ftr(7));  
    System.out.println(ftr(11));  
}
```

NB:

Penyelesaian masalah ini dapat dilakukan secara rekursif



UNIVERSITAS
ISLAM
INDONESIA



UNIVERSITAS
ISLAM
INDONESIA

FUNDAMEN PENGEMBANGAN APLIKASI

Pertemuan 5

Terima Kasih