

JOBSHEET 13

PRAKTIKUM DASAR PEMROGRAMAN

Oleh: Maharani Wirawan (2341760111)

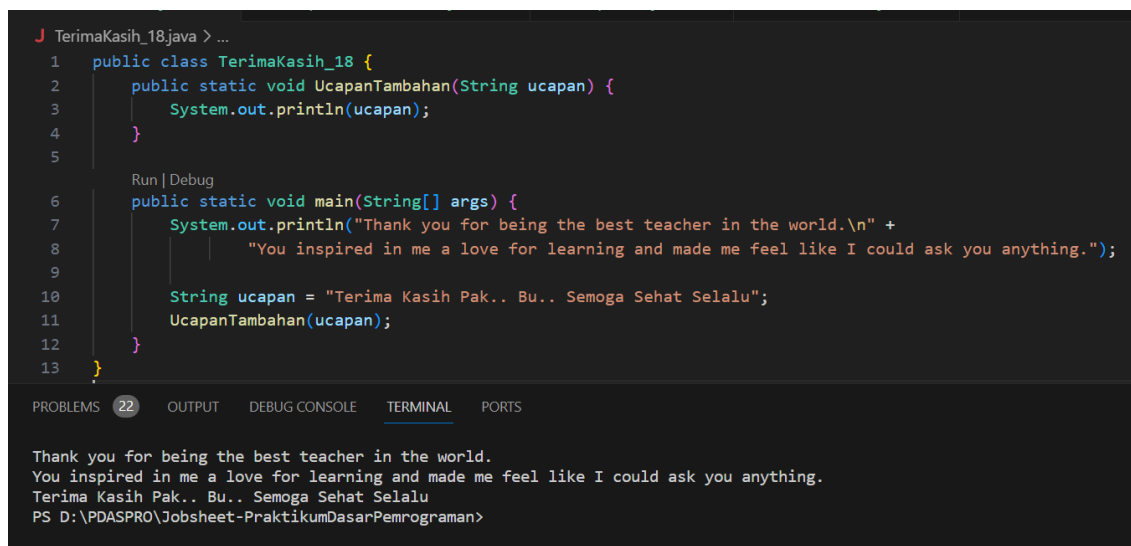
• PERCOBAAN 1

1. Apakah fungsi tanpa parameter selalu digunakan untuk fungsi “void” saja?

Tidak, fungsi tanpa parameter tidak selalu digunakan untuk fungsi "void" saja. Fungsi tersebut bisa mengembalikan nilai (return) tanpa menerima parameter.

2. Apakah bisa kalimat “Thank you for.....dst” dituliskan tanpa fungsi UcapanTerimaKasih? modifikasi kode program tersebut untuk dapat menampilkan suatu kalimat tanpa dan dengan menggunakan fungsi!

Bisa, kalimat "Thank you for..." dapat dituliskan langsung di dalam fungsi main tanpa menggunakan fungsi UcapanTerimaKasih. Dengan modifikasi:



```
J TerimaKasih_18.java > ...
1 public class TerimaKasih_18 {
2     public static void UcapanTambahan(String ucapan) {
3         System.out.println(ucapan);
4     }
5
6     Run | Debug
7     public static void main(String[] args) {
8         System.out.println("Thank you for being the best teacher in the world.\n" +
9             "You inspired in me a love for learning and made me feel like I could ask you anything.");
10
11         String ucapan = "Terima Kasih Pak.. Bu.. Semoga Sehat Selalu";
12         UcapanTambahan(ucapan);
13     }
14 }
```

PROBLEMS 22 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Thank you for being the best teacher in the world.
You inspired in me a love for learning and made me feel like I could ask you anything.
Terima Kasih Pak.. Bu.. Semoga Sehat Selalu
PS D:\PDASPRO\Jobsheet-PraktikumDasarPemrograman>

3. Apakah keuntungan menggunakan fungsi di dalam program?

Menggunakan fungsi di dalam program membawa keuntungan dengan memisahkan kode menjadi bagian-bagian kecil yang lebih mudah dimengerti. Fungsi memudahkan uji coba, mengurangi kekeliruan, dan memungkinkan penggunaan kembali kode. Selain itu, penggunaan fungsi membuat kode terorganisir dengan baik, mempermudah dalam pemahaman dan pemeliharaan. Jadi, penggunaan fungsi membantu menjadikan kode lebih efisien dan terkelola.

- **PERCOBAAN 2**

- 1. Apakah kegunaan suatu parameter di dalam fungsi?**

Parameter adalah variabel yang digunakan dalam deklarasi fungsi atau metode untuk menerima nilai yang akan dioperasikan oleh fungsi tersebut. Kegunaan utama parameter adalah memungkinkan fungsi menerima input atau argumen yang berbeda setiap kali dipanggil, membuat fungsi lebih fleksibel dan dapat digunakan dengan berbagai nilai.

- 2. Apakah parameter sama dengan variabel? jelaskan!**

Ya, parameter adalah jenis variabel yang digunakan untuk menerima nilai saat fungsi dipanggil. Namun, parameter hanya memiliki cakupan (scope) di dalam tubuh fungsi tempat mereka dideklarasikan.

- 3. Apakah parameter di dalam Bahasa pemrograman Java hanya digunakan untuk melewati data input saja? Bagaimana dengan data output?**

Parameter di Java tidak hanya digunakan untuk menerima data input, tetapi juga untuk mengembalikan data output. Sebagai contoh, nilai yang dikembalikan oleh suatu fungsi dapat ditentukan melalui tipe data pengembalian fungsi (misalnya, `int`, `double`, `String`, dll.). Selain itu, parameter juga digunakan untuk memberikan fleksibilitas dalam penggunaan fungsi, memungkinkan fungsi tersebut bekerja dengan nilai yang diberikan saat dipanggil. Dengan parameter, fungsi dapat diimplementasikan untuk melakukan tugas tertentu dengan nilai yang bervariasi tergantung pada konteks pemanggilannya.

- **PERCOBAAN 3**

- 1. Jelaskan mengenai, kapan suatu fungsi membutuhkan nilai kembalian (return)?**

Suatu fungsi membutuhkan nilai kembalian (`return`) ketika hasil dari operasi di dalam fungsi perlu dikirim atau dikembalikan ke pemanggilnya.

- 2. Apakah bisa ditambahkan statement `System.out.println` di dalam fungsi dengan nilai kembalian? Apa pengaruh nya?**

Ya, bisa ditambahkan statement `System.out.println` di dalam fungsi dengan nilai kembalian. Pengaruhnya adalah statement tersebut akan dieksekusi saat fungsi dipanggil, tetapi tidak mempengaruhi nilai kembalian fungsi. Fungsi tetap mengembalikan nilai yang telah ditentukan oleh pernyataan `return`.

3. Apakah bisa fungsi tanpa nilai kembalian di dalam fungsi main dipanggil tanpa dilewatkan ke variabel? seperti pada percobaan 1? Jelaskan!

Ya, fungsi tanpa nilai kembalian (void) di dalam fungsi `main` dapat dipanggil tanpa dilewatkan ke variabel. Contohnya, pada percobaan 3, pemanggilan `PenerimaUcapan()` di fungsi `main` tidak perlu menyimpan nilai kembalian ke dalam variabel, karena fungsi tersebut memiliki tipe `String` dan mengembalikan nilai langsung digunakan dalam operasi `System.out.println`. Pemanggilan `PenerimaUcapan()` dapat dilakukan seperti yang ditunjukkan pada kode Anda, yaitu tanpa menyimpan nilai kembalian ke dalam variabel.

• **PERCOBAAN 4**

1. Berdasarkan uji coba pada percobaan 4, manakah fungsi yang akan eksekusi pertama kali? Jelaskan!

Fungsi yang akan dieksekusi pertama kali adalah fungsi `main` karena program Java selalu dimulai dari fungsi `main`. Pada program di atas, `UcapanTerimaKasih()` dipanggil dari dalam fungsi `main`, sehingga eksekusi dimulai dari situ.

2. Manakah tata cara penulisan fungsi benar di dalam class? Di atas fungsi main atukah dibawah fungsi main? Jelaskan!

Dalam Java, tata cara penulisan fungsi adalah mendeklarasikan fungsi-fungsi di dalam class. Secara umum, adalah menempatkan fungsi `main` di atas fungsi-fungsi lainnya. Ini tidak mempengaruhi program karena JVM (Java Virtual Machine) akan mencari fungsi `main` sebagai titik awal eksekusi program.

3. Modifikasi program di atas dengan menambahkan fungsi UcapanTambahan() dengan input parameter String. Fungsi UcapanTambahan() berisi ucapan tambahan yang ingin disampaikan kepada penerima ucapan.

```
24 public class UcapanTerimaKasih_18 {
25
26     public static String PenerimaUcapan() {
27         Scanner sc = new Scanner(System.in);
28         System.out.println(x:"Tuliskan NAMA orang yang ingin Anda beri ucapan:");
29         String namaOrang = sc.nextLine();
30         sc.close();
31         return namaOrang;
32     }
33     public static void UcapanTerimaKasih() {
34         String nama = PenerimaUcapan();
35         System.out.println("Thank you " + nama + " for being the best teacher in the world.\n" +
36             "You inspired in me a love for learning and made me feel like I could ask you anything.");
37         UcapanTambahan(ucapan:"Semoga sukses selalu!");
38     }
39     public static void UcapanTambahan(String ucapan) {
40         System.out.println(ucapan);
41     }
42
43     public static void main(String[] args) {
44         UcapanTerimaKasih();
45     }
46 }
47
48 Run | Debug
49
50 public static void main(String[] args) {
51     UcapanTerimaKasih();
52 }
53
54 PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL PORTS
55
56 Tuliskan NAMA orang yang ingin Anda beri ucapan:
57 RANI
58 Thank you RANI for being the best teacher in the world.
59 You inspired in me a love for learning and made me feel like I could ask you anything.
60 Semoga sukses selalu!
61 PS D:\PDASPRO\Jobsheet-PraktikumDasarPemrograman> |
```

• PERCOBAAN 5

1. Jelaskan mengapa penulisan parameter di praktikum 5 di tulis dengan int... a!

Penulisan `int... a` disebut sebagai varargs (variable-length argument) dalam Java. Ini memungkinkan fungsi menerima jumlah argumen variabel atau parameter yang dapat disesuaikan oleh pemanggil fungsi. Dengan menggunakan varargs, kita dapat memanggil fungsi dengan jumlah argumen yang bervariasi, bahkan tidak memberikan argumen sama sekali.

2. Sebutkan kegunaan varargs dalam implementasi kode program untuk menyelesaikan permasalahan dalam dunia nyata! (minimal 3)

- 1) **Fleksibilitas dalam Pemanggilan Fungsi:** Varargs memungkinkan pemanggilan fungsi dengan jumlah argumen yang bervariasi, memberikan fleksibilitas kepada pengguna fungsi.
- 2) **Mengatasi Keterbatasan Jumlah Parameter Tetap:** Varargs dapat digunakan untuk mengatasi keterbatasan jumlah parameter tetap pada fungsi, sehingga tidak perlu membuat banyak versi fungsi dengan jumlah parameter yang berbeda.

- 3) Meningkatkan Keterbacaan Kode: Varargs dapat meningkatkan keterbacaan kode karena menghilangkan kebutuhan untuk mendeklarasikan banyak parameter tetap, terutama ketika jumlahnya tidak dapat diprediksi.

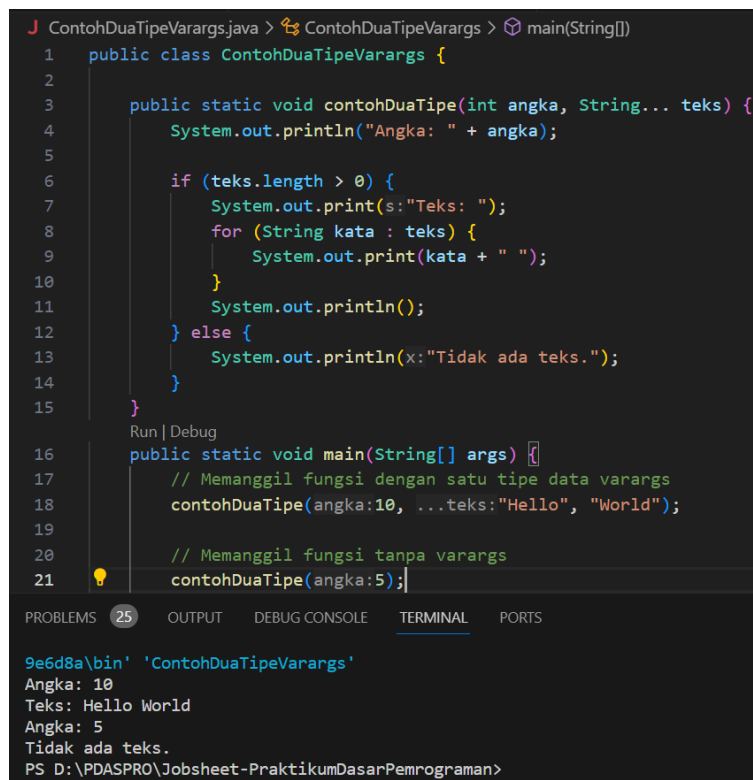
3. Bisakah kita menggunakan dua tipe data varargs dalam satu fungsi? Berikan contohnya!

Tidak, satu fungsi hanya dapat memiliki satu varargs. Jika kita mencoba menggunakan dua varargs atau lebih, akan terjadi kesalahan kompilasi.

```
public static void contohVarargs(int... a, String... b) {  
    // Ini akan menghasilkan kesalahan kompilasi  
    // Kita hanya dapat memiliki satu varargs dalam satu fungsi  
}
```

- Jika kita memerlukan dua tipe data, kita bisa menggunakan parameter biasa dan varargs sekaligus:

```
public static void contohDuaTipe(int angka, String... teks) {  
    // Implementasi fungsi dengan parameter dan varargs  
}
```



```
J ContohDuaTipeVarargs.java > ContohDuaTipeVarargs > main(String[])  
1 public class ContohDuaTipeVarargs {  
2  
3     public static void contohDuaTipe(int angka, String... teks) {  
4         System.out.println("Angka: " + angka);  
5  
6         if (teks.length > 0) {  
7             System.out.print(s:"Teks: ");  
8             for (String kata : teks) {  
9                 System.out.print(kata + " ");  
10            }  
11            System.out.println();  
12        } else {  
13            System.out.println(x:"Tidak ada teks.");  
14        }  
15    }  
16  
17    public static void main(String[] args) {  
18        // Memanggil fungsi dengan satu tipe data varargs  
19        contohDuaTipe(angka:10, ...teks:"Hello", "World");  
20  
21        // Memanggil fungsi tanpa varargs  
22        contohDuaTipe(angka:5);  
23    }  
24 }  
Run | Debug  
9e6d8a\bin 'ContohDuaTipeVarargs'  
Angka: 10  
Teks: Hello World  
Angka: 5  
Tidak ada teks.  
PS D:\PDASPRO\Jobsheet-PraktikumDasarPemrograman>
```