

Jobsheet 11 Praktikum Daspro

Nama: Dafa Naufal Rabbani

Kelas/No.Absen: TI-1G/05

NIM: 254107020086

Jawaban Percobaan 1:

```
Jobsheet11 > J Kafe05.java > kafe05 > Menu()
1 package Jobsheet11;
2
3 public class kafe05 {
4     public static void Menu() {
5         System.out.println("==== MENU RESTO KAFE ====");
6         System.out.println("1. Kopi Hitam - Rp 15,000");
7         System.out.println("2. Cappuccino - Rp 20,000");
8         System.out.println("3. Latte - Rp 22,000");
9         System.out.println("4. Teh Tarik - Rp 12,000");
10        System.out.println("5. Roti Bakar - Rp 10,000");
11        System.out.println("6. Mie bakar - Rp 18,000");
12        System.out.println("=====");
13        System.out.println("Silahkan pilih menu yang Anda inginkan.");
14    }
15
16    Run | Debug
17    public static void main(String[] args) {
18        Menu();
19    }
20
21 }
```

1. Apakah selalu harus:

- Tidak, fungsi tanpa parameter tidak harus bertipe void karena fungsi tersebut tidak menerima input, tetapi boleh mengembalikan nilai.

2. Apakah bisa:

- Bisa, semua perintah System.out.println() dapat langsung ditulis di dalam main().

3. Keuntungan:

- a) Kode lebih rapi dan mudah dibaca
- b) Menghindari duplikasi kode
- c) Memudahkan perubahan
- d) Struktur program lebih modular

4. Alur singkat:

- 1) Program Java dijalankan yang dimulai dari fungsi main()
- 2) Di dalam main(), program menemukan perintah Menu();
- 3) Program memanggil fungsi Menu() dan langsung pindah untuk mengeksekusi isi fungsi tersebut.
- 4) Di dalam Menu(), semua System.out.println() dieksekusi satu per satu.
- 5) Setelah selesai menampilkan seluruh daftar menu, fungsi Menu() selesai.
- 6) Program kembali ke fungsi main()

Jawaban Percobaan 2:

```
Jobsheet11 > J Kafe05.java > ...
1 package Jobsheet11;
2
3 public class kafe05 {
4     public static void Menu(String namaPelanggan, boolean isMember) {
5         System.out.println("Selamat datang, " + namaPelanggan + "!");
6
7         if(isMember) {
8             System.out.println("Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");
9         }
10
11        System.out.println("==== MENU RESTO KAFE ====");
12        System.out.println("1. Kopi Hitam - Rp 15,000");
13        System.out.println("2. Cappuccino - Rp 20,000");
14        System.out.println("3. Latte - Rp 22,000");
15        System.out.println("4. Teh Tarik - Rp 12,000");
16        System.out.println("5. Roti Bakar - Rp 10,000");
17        System.out.println("6. Mie bakar - Rp 18,000");
18        System.out.println("=====");
19        System.out.println("Silahkan pilih menu yang Anda inginkan.");
20    }
21
22 Run | Debug
23 public static void main(String[] args) {
24     Menu("Andi", true);
25 }
26
27 }
```

1. Kegunaan:

- Parameter digunakan untuk mengirimkan data dari luar fungsi ke dalam fungsi, dengan parameter fungsi bisa menerima nilai yang berbeda-beda setiap kali dipanggil.

2. Mengapa:

- Karena namaPelanggan digunakan untuk menampilkan nama orang yang sedang dilayani dan isMember digunakan untuk menentukan apakah pelanggan mendapat diskon atau tidak.

3. Apakah sama:

- Mirip, tetapi tidak sama karena parameter adalah variabel khusus yang digunakan untuk menerima nilai dari pemanggil fungsi sedangkan variabel biasa adalah nilai yang dideklarasikan di dalam fungsi untuk keperluan internal.

4. Cara kerja parameter isMember dan perbedaan output:

- isMember bertipe boolean, jadi hanya memiliki dua nilai yaitu true atau false:
 - a) Jika true program menampilkan:
"Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!"
 - b) Jika false program menampilkan:
Bagian diskon tidak ditampilkan.

5. Apa yang akan terjadi:

- Akan terjadi error karena fungsi Menu() wajib menerima dua parameter yaitu Menu(String namaPelanggan, boolean isMember), jika dipanggil Menu(); maka java akan menampilkan eror\

6. Modifikasi:

```
Jobsheet11 > J Kafe05.java > kafe05 > main(String[])
1 package Jobsheet11;
2
3 public class kafe05 {
4     public static void Menu(String namaPelanggan, boolean isMember, String kodePromo) {
5         System.out.println("Selamat datang, " + namaPelanggan + "!");
6
7         if(isMember) {
8             System.out.println("Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");
9         }
10
11         // Promo
12         if(kodePromo.equals("DISKON50")) {
13             System.out.println("Kode promo valid! Anda mendapatkan diskon 50%");
14         } else if(kodePromo.equals("DISKON30")) {
15             System.out.println("Kode promo valid! Anda mendapatkan diskon 30%");
16         } else {
17             System.out.println("Kode promo invalid.");
18         }
19
20         System.out.println("==== MENU RESTO KAFE ====");
21         System.out.println("1. Kopi Hitam - Rp 15,000");
22         System.out.println("2. Cappucino - Rp 20,000");
23         System.out.println("3. Latte - Rp 22,000");
24         System.out.println("4. Teh Tarik - Rp 12,000");
25         System.out.println("5. Roti Bakar - Rp 10,000");
26         System.out.println("6. Mie bakar - Rp 18,000");
27         System.out.println("=====");
28         System.out.println("silahkan pilih menu yang Anda inginkan.");
29     }
30
31     Run | Debug
32     public static void main(String[] args) {
33         Menu("Andi", true, "DISKON50");
34     }
35 }
```

7. Perintah pemanggilan fungsi:

- Menu("Budi", true, "DISKON30");

8. Apakah lebih mudah:

- Ya, jauh lebih mudah dibaca dan dikembangkan karena:
 - Tidak perlu mengubah isi fungsi setiap kali nama pelanggan berubah
 - Fungsi menjadi fleksibel
 - Mengurangi duplikasi
 - Lebih rapi dan profesional

Jawaban Percobaan 3:

```
Jobsheet11 > J Kafe05.java > ...
5  public class kafe05 {
6      public static void Menu(String namaPelanggan, boolean isMember, String kodePromo) {
7          if(kodePromo.equals("DISKON30")) {
8              System.out.println("Kode promo valid! Anda mendapatkan diskon 30%!");
9          } else {
10             System.out.println("Kode promo invalid.");
11         }
12
13         System.out.println("\n==== MENU RESTO KAFE ====");
14         System.out.println("1. Kopi Hitam - Rp 15,000");
15         System.out.println("2. Cappuccino - Rp 20,000");
16         System.out.println("3. Latte - Rp 22,000");
17         System.out.println("4. Teh Tarik - Rp 12,000");
18         System.out.println("5. Roti Bakar - Rp 10,000");
19         System.out.println("6. Mie Bakar - Rp 18,000");
20         System.out.println("=====");
21     }
22
23     public static int hitungTotalHargaNoPresensi(int pilihanMenu, int banyakItem) {
24         int[] hargaItems = {15000, 20000, 22000, 12000, 10000, 18000};
25         int hargaTotal = hargaItems[pilihanMenu - 1] * banyakItem;
26         return hargaTotal;
27     }
28
29
30
31     Run | Debug
32     public static void main(String[] args) {
33
34         Scanner sc = new Scanner(System.in);
35
36         Menu("Andi", true, "DISKON50");
37
38         System.out.print("\nMasukkan nomor menu yang ingin Anda pesan: ");
39         int pilihanMenu = sc.nextInt();
40
41         System.out.print("Masukkan jumlah item yang ingin dipesan: ");
42         int banyakItem = sc.nextInt();
43
44
45
46
47
48
49 }
```

1. Kapan:

- Fungsi membutuhkan nilai kembalian ketika hasil perhitungannya akan digunakan lagi oleh bagian program lain dan jika fungsi hanya melakukan tampilan atau proses tanpa butuh nilai balik, maka tidak memerlukan return value.

2. Tipe data nilai kembalian & parameter fungsi:

A. Tipe data nilai kembalian:

- 1) int: Karena fungsi mengembalikan total harga dalam bentuk angka integer.

B. Dua parameter yang digunakan:

- 1) int pilihanMenu: Angka menu yang dipilih pengguna (1 sampai 6).
- 2) int banyakItem: Jumlah pesanan item untuk menu tersebut.

Jawaban Percobaan 4:

```
Jobsheet11 > J PengunjungCafe05.java > ...
1 package Jobsheet11;
2
3 public class PengunjungCafe05 {
4     static void daftarPengunjung(String... namaPengunjung) {
5         System.out.println("Daftar Nama Pengunjung :");
6         for (int i = 0; i < namaPengunjung.length; i++) {
7             System.out.println("- " + namaPengunjung[i]);
8         }
9     }
10
11    Run | Debug
12    public static void main(String[] args) {
13        daftarPengunjung("Ali", "Budi", "Citra");
14        daftarPengunjung("Andi");
15        daftarPengunjung("Doni", "Eti", "Fahmi", "Galih");
16    }
17 }
```

1. Mengapa:

- Penulisan String... namaPengunjung digunakan karena varargs (variable arguments) memungkinkan satu method menerima jumlah argumen yang tidak terbatas, tanpa harus membuat array secara manual.
Artinya, method dapat dipanggil seperti:
 - a. daftarPengunjung("Ali", "Budi")
 - b. daftarPengunjung("Andi")
 - c. daftarPengunjung("Doni", "Eti", "Fahmi")

2. Modifikasi:

```
Jobsheet11 > J PengunjungCafe05.java > ...
1 package Jobsheet11;
2
3 public class PengunjungCafe05 {
4     static void daftarPengunjung(String... namaPengunjung) {
5         System.out.println("Daftar Nama Pengunjung :");
6         for (String nama : namaPengunjung) {
7             System.out.println("- " + nama);
8         }
9     }
10
11    Run | Debug
12    public static void main(String[] args) {
13        daftarPengunjung("Ali", "Budi", "Citra");
14        daftarPengunjung("Andi");
15        daftarPengunjung("Doni", "Eti", "Fahmi", "Galih");
16    }
17 }
```

3. Apakah bisa:

- Tidak bisa, kare java hanya memperbolehkan satu varargs dalam satu method, dan harus berada di posisi terakhir dalam parameter list.
4. Apa yang terjadi:
- Hasil yang keluar:
 - a. Program tetap berjalan
 - b. Tidak error compile-time maupun runtime
 - c. Varargs menerima array kosong
 - d. Loop tidak mencetak nama apa pun, tetapi header tetap diprint