

Jobsheet 11

Nama : Lindhu Nuril Rahmatdanto

NIM : 254107020216

Kelas : 1G

No.Absen : 17

Pertanyaan

1. Apakah fungsi tanpa parameter selalu harus bertipe void?

*Team Teaching Dasar Pemrograman 2025
Politeknik Negeri Malang*

Dasar Pemrograman 2025



2. Apakah daftar menu pada program kafe dapat ditampilkan tanpa menggunakan fungsi **Menu()**? Modifikasi kode program tersebut untuk dapat menampilkan daftar menu tanpa menggunakan fungsi!
3. Jelaskan keuntungan menggunakan fungsi Menu() dibandingkan menulis semua perintah penampilan menu langsung di dalam fungsi main.
4. Uraikan secara singkat alur eksekusi program ketika fungsi Menu() dipanggil dari main (mulai dari program dijalankan sampai daftar menu tampil di layar).

Jawab :

- 1.Tidak selalu,tergantung kebutuhan programmer.Apabila programmer menginginkan nilai yang di kerjakan suatu fungsi untuk dikembalikan dan ditampilkan di fungsi “main” program tidak memakai parameter void melainkan return dan sebaliknya.

```
1 package Jobsheet11;
2
3 public class Kafe17 {
4     public static void Menu (){
5     }
6     public static void main(String[] args) {
7         System.out.println(x: "==== Menu Resto Kafe ===");
8         System.out.println(x: "1. Kopi Hitam - Rp 15,000");
9         System.out.println(x: "2. Cappuccino - Rp 20,000");
10        System.out.println(x: "3. Latte - Rp 22,000");
11        System.out.println(x: "4. Teh Tarik - Rp 12,000");
12        System.out.println(x: "5. Roti Bakar - Rp 10,000");
13        System.out.println(x: "6. Mie Goreng - Rp 18,000");
14        System.out.println(x: "=====");
15    }
16 }
17
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\tempe\Documents\java\MatKul> c:; cd 'c:\Users\tempe\Document va.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\tempe\A b3804131bb7bc89641e4\redhat.java\jdt_ws\MatKul_9ab8b29b\bin' 'Jobsheet11.
==== Menu Resto Kafe ===
1. Kopi Hitam - Rp 15,000
2. Cappuccino - Rp 20,000
3. Latte - Rp 22,000
4. Teh Tarik - Rp 12,000
5. Roti Bakar - Rp 10,000
6. Mie Goreng - Rp 18,000
=====
```

2. Tanpa memanggil fungsi yang ada program akan mustahil menampilkan output yang sama seperti yang ada di fungsi menu();,kecuali memindahkan seluruh isi program fungsi menu ke fungsi main secara manual.

3.Dengan menggunakan fungsi menu programmer dapat mempersingkat dan mendefinisikan setiap kerja fungsi fungsi yang ada dalam programnya (apabila programnya panjang dan rumit) sehingga lebih mudah dimengerti dan efisien.

4.Dalam program diatas,fungsi yang akan dijalankan pertama kali adalah fungsi main,baru setelah itu komputer akan membaca apa instruksi dari fungsi main ; membaca fungsi menu();.Di dalam fungsi menu terdapat tampilan ragam menu yang ditulis di programnya.

-
1. Apakah kegunaan parameter di dalam fungsi?
 2. Jelaskan mengapa pada percobaan ini fungsi **Menu()** menggunakan parameter **namaPelanggan** dan **isMember**?
 3. Apakah parameter sama dengan variabel? Jelaskan.

*Team Teaching Dasar Pemrograman 2025
Politeknik Negeri Malang*

Dasar Pemrograman 2025



4. Jelaskan bagaimana cara kerja parameter **isMember** pada fungsi **Menu()**. Apa perbedaan output ketika **isMember** bernilai true dan ketika false?
 5. Apa yang akan terjadi jika memanggil fungsi **Menu()** tanpa menyertakan parameter **namaPelanggan** dan **isMember**?
 6. Modifikasi kode di atas dengan menambahkan parameter baru **kodePromo (String)**. Jika **kodePromo** adalah "DISKON50", tampilkan berikan diskon 50%. Jika **kodePromo** adalah "DISKON30", tampilkan berikan diskon 30%. Jika tidak ada kode promo yang berlaku, tampilkan kode invalid.
 7. Berdasarkan fungsi **Menu()** di atas, jika nama pelanggan adalah "Budi", pelanggan tersebut member, dan menggunakan kode promo "DISKON30", tuliskan satu baris perintah pemanggilan fungsi menu yang benar.
 8. Menurut Anda, apakah penggunaan parameter **namaPelanggan** dan **isMember** pada fungsi **Menu()** membuat program lebih mudah dibaca dan dikembangkan dibandingkan jika nilai-nilai tersebut ditulis langsung di dalam fungsi tanpa parameter? Jelaskan alasan Anda.
 9. **Commit** dan **push** hasil modifikasi Anda ke **Github** dengan pesan "**Modifikasi Percobaan 2**"
-

Jawab :

1. Kegunaan parameter dalam fungsi adalah agar fungsi tersebut dapat menerima data dari luar agar kerjanya fleksibel dan tidak terkunci pada satu nilai tetap.

2. Agar bisa menjalankan 4 row program di bawahnya. Parameter **namaPelanggan** berfungsi untuk mengucapkan selamat datang, sedangkan parameter **isMember** berfungsi untuk menjalankan program "if" yang apabila true maka akan dijalankan apabila tidak maka akan di skip.

3. Tidak sama secara umum, tapi parameter adalah variabel "khusus" karena terletak di dalam kurung dan harus dipanggil dari luar. Seperti di dalam konteks program ini String **namaPelanggan** dan boolean **isMember** harus dipanggil dari fungsi **main** tepatnya di dalam kurung fungsi **menu**.

4.Cara kerja parameter isMember bergantung pada nilai boolean yang ditaruh di dalam kurung menu().Apabila nilainya true maka dalam fungsi menu() program if akan dilaksanakan,apabila nilai yang ada pada menu() adalah false maka program if akan di skip.

5.Program akan error karena sebagian program membutuhkan nilai dari namaPelanggan dan isMember.Seperti dalam konteks program ini apabila namaPelanggan dihilangkan otomatis program eror karena untuk ucapan selamat datang tidak ada variabel dan nilai yang diterima.

6.

```
package Jobsheet11;

public class Kafe17 {
    public static void Menu (String namaPelanggan,boolean isMember,String kodePromo){
        System.out.println("Selamat datang " + namaPelanggan + " !");
        if (isMember) {
            System.out.println("Anda adalah member,dapatkan diskon 10% untuk setiap pembelian!");
        }
        if (kodePromo.equals(anObject: "DISKON50")) {
            System.out.println("Selamat anda mendapatkan diskon sebesar 50%");
        }else if (kodePromo.equals(anObject: "DISKON30")) {
            System.out.println("Selamat anda mendapatkan diskon sebesar 30%");
        }else {
            System.out.println("Kode promo invalid");
        }
        System.out.println("==== Menu Resto Kafe ===");
        System.out.println("1. Kopi Hitam - Rp 15,000");
        System.out.println("2. Cappuccino - Rp 20,000");
        System.out.println("3. Latte - Rp 22,000");
        System.out.println("4. Teh Tarik - Rp 12,000");
        System.out.println("5. Roti Bakar - Rp 10,000");
        System.out.println("6. Mie Goreng - Rp 18,000");
        System.out.println("=====");
    }
}

Run | Debug
public static void main(String[] args) {
    Menu(namaPelanggan: "Budi",isMember: true,kodePromo: "DISKON30");
}
```

7.

```
Run | Debug
public static void main(String[] args) {
    Menu(namaPelanggan: "Budi",isMember: true,kodePromo: "DISKON30");
```

8.Iya,karena menurut saya function ini membuat kode program jadi ringkas dan efisien sehingga mudah dibaca dan dimengerti.Juga dengan adanya function ini kode jadi terlihat lebih struktural dan rapi.

9.

NB : Maaf sebelumnya ibu,bahwa commit untuk percobaan dan modifikasi dalam program ini telah saya lakukan hanya saja letaknya berada di file jobsheet 10.

5 days ago		Percobaan 1		1
				2
yesterday		Percobaan 3		3
				4
5 days ago		Percobaan 1		5
3 days ago		Modifikasi Percobaan 2		6
yesterday		Percobaan 3		7
4 days ago		Percobaan 2		8
				9
				10
				11
3 days ago		Modifikasi Percobaan 2		12
				13
				14
				15
				16
				17
				18
5 days ago		Percobaan 1		19
				20
				21
				22
				23
				24
				25
				26

			26
			27
			28
			29
			30
			31
			32
			33
			34
			35
			36
			37
			38
			39
			40
			41
			42
			43
			44

Ini adalah bukti bahwa saya telah meng commit percobaan sesuai dengan intruksi yang ibu dosen berikan. Sebelumnya saya mohon maaf ibu.

Pertanyaan

1. Jelaskan secara singkat kapan suatu fungsi membutuhkan nilai kembalian (*return value*) dan kapan fungsi tidak perlu mengembalikan nilai. Berikan minimal satu contoh dari program kafe pada Percobaan 3 untuk masing-masing kasus.
2. Fungsi **hitungTotalHargaNoPresensi** saat ini mengembalikan total harga berdasarkan **pilihanMenu** dan **jumlahPesanan**. Sebutkan tipe data nilai kembalian dan dua buah parameter yang digunakan fungsi tersebut. Jelaskan arti masing-masing parameter dalam konteks program kafe.
3. Modifikasi kode di atas sehingga fungsi **hitungTotalHargaNoPresensi** dapat menerima **kodePromo**. Jika **kodePromo** adalah "DISKON50", maka mendapat diskon 50% dari **totalHarga** dan tampilkan diskon. Jika **kodePromo** adalah "DISKON30", maka mendapat diskon 30% dari **totalHarga** dan tampilkan diskon. Jika tidak ada kode promo yang berlaku, tampilkan kode invalid dan tidak ada pengurangan total harga **totalHarga**.
4. Modifikasi kode di atas sehingga bisa memilih beberapa jenis menu berbeda serta menampilkan total keseluruhan pesanan. Bagaimana memodifikasi program sehingga pengguna dapat: memesan **lebih dari satu jenis menu** (misalnya menu 1 dan 3 sekaligus), dan menampilkan **total keseluruhan** pesanan (gabungan dari semua jenis menu)?
5. **Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 3”**

1. Suatu program butuh return value apabila tipe data yang dipasang sebelum nama fungsi adalah tipe data selain void. Contoh seperti di fungsi hitungTotalHarga17 tipe data yang dipasang disana adalah int dan bukan void. Ini berarti tipe data akan dikembalikan dalam bentuk Int. Apabila return di hapus dalam program, yang terjadi adalah error karena tipe data yang dipasang (int) menuntut adanya value yang dikembalikan.

2. Tipe data yang dipakai dalam function tersebut adalah integer. Artinya valuenya akan bertipe integer. Dua parameter yang ada didalam kurung berfungsi untuk menerima nilai dari luar. Contoh yang ada pada fungsi main

```
J
Run | Debug
public static void main(String[] args) {
    Menu(namaPelanggan: "Budi",isMember: true,kodePromo: "DISKON30");
    Scanner sc = new Scanner(System.in);
    System.out.print(s: "Masukkan nomor menu yang ingin anda pesan :");
    int pilihanMenu = sc.nextInt();
    System.out.print(s: "Masukkan jumlah item yang ingin dipesan :");
    int banyakItem = sc.nextInt();

    int totalHarga = hitungTotalHarga17(pilihanMenu, banyakItem);

    System.out.println("Total harga untuk pesanan anda : " +totalHarga);

}
```

Dalam baris kedua dari terakhir terlihat ada variabel pilihanMenu dan banyakItem yang merupakan parameter dari fuction hitungTotalHarga17

3.

```
J
}
public static int hitungTotalHarga17(int pilihanMenu,int banyakItem,String kodePromo){
    int [] hargaItems = {15000,20000,22000,12000,10000,18000};

    int hargaTotal = hargaItems [pilihanMenu - 1] *banyakItem;
    if (kodePromo.equals(anObject: "DISKON50")) {
        hargaTotal -= hargaTotal * 50 /100;
    }else if (kodePromo.equals(anObject: "DISKON30")) {
        hargaTotal -= hargaTotal * 30 /100;
    }
    return hargaTotal;
}
Run | Debug
public static void main(String[] args) {
    Menu(namaPelanggan: "Budi",isMember: true,kodePromo: "DISKON30");
    Scanner sc = new Scanner(System.in);
    System.out.print(s: "Masukkan nomor menu yang ingin anda pesan :");
    int pilihanMenu = sc.nextInt();
    System.out.print(s: "Masukkan jumlah item yang ingin dipesan :");
    int banyakItem = sc.nextInt();

    int totalHarga = hitungTotalHarga17(pilihanMenu, banyakItem,kodePromo: "DISKON30");

    System.out.println("Total harga untuk pesanan anda adalah : "+ totalHarga);

}
```

4.

```
        return hargaTotal;
    }

Run | Debug
public static void main(String[] args) {
    Menu(namaPelanggan: "Budi",isMember: true,kodePromo: "DISKON30");
    Scanner sc = new Scanner(System.in);
    int totalHargaFinal = 0;
    while (true) {

        System.out.print(s: "Masukkan nomor menu yang ingin anda pesan :");
        int pilihanMenu = sc.nextInt();
        System.out.print(s: "Masukkan jumlah item yang ingin dipesan :");
        int banyakItem = sc.nextInt();
        sc.nextLine();

        int totalHarga = hitungTotalHarga17(pilihanMenu, banyakItem,kodePromo: "DISKON30");

        totalHargaFinal += totalHarga;
        System.out.print(s: "Apa ada yang lain : (y/n) :");
        String opsi = sc.nextLine();

        if (opsi.equalsIgnoreCase(anotherString: "n")) {
            break;
        }

    }
    System.out.println("Berikut adalah total harga yang harus anda bayar : "+totalHargaFinal);
}
```

Pertanyaan

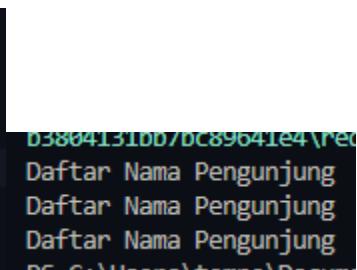
1. Jelaskan mengapa penulisan parameter di praktikum 4 ditulis dengan **String...** **namaPengunjung!**
 2. Modifikasi method **daftarPengunjung** menggunakan **for-each** loop.
 3. Bisakah menggunakan dua tipe data varargs dalam satu fungsi? Jelaskan jawaban Anda berdasarkan aturan varargs di Java, dan berikan contohnya!
 4. Jelaskan apa yang terjadi jika fungsi **daftarPengunjung** dipanggil tanpa argumen. Apakah program akan error saat kompilasi, error saat dijalankan, atau tetap berjalan? Jika tetap berjalan, bagaimana output yang dihasilkan?
 5. **Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 4”**
- 1.Karena yang dipakai adalah Varargs,yang memungkinkan pengguna menginput data sebanyak-banyaknya dan jumlah argumen yang tidak tetap.

```
public class PengunjungCafe17 {
    static void daftarPengunjung(String... namaPengunjung){
        System.out.println("Daftar Nama Pengunjung");
        for (String nama : namaPengunjung){
            System.out.println("- " + namaPengunjung);
        }
    }
}
```

3. Tidak bisa karena menurut peraturan setiap fungsi hanya diperkenankan memiliki 1 varargs. Apabila tidak begitu maka java akan bingung ketika ada 2 tipe data dan ada nilai masuk, java tidak bisa memilah mana nilai yang termasuk tipe data yang mana. Contoh ketika ada tipe data integer dan string, java akan bingung ketika ada 2 nilai yang masuk. Yang pertama adalah “Dua” dan yang kedua adalah 1. Ini akan menimbulkan error

4. Program akan tetap berjalan tanpa adanya nama-nama disana. Yang akan terprint hanya kalimat Daftar Nama Pengunjung

```
Run | Debug
public static void main(String[] args) {
    daftarPengunjung();
    daftarPengunjung();
    daftarPengunjung();
}
```



D3804131007DC89641e4\reco
Daftar Nama Pengunjung
Daftar Nama Pengunjung
Daftar Nama Pengunjung
D:\...\Reco\reco\src\main\java\reco\