

**NAMA : ALFATITAH ALIFIA PUTRI**

**PRODI/ASEN : SIB 1E / 04**

**NIM : 254107060143**

## **PERCOBAAN 1**

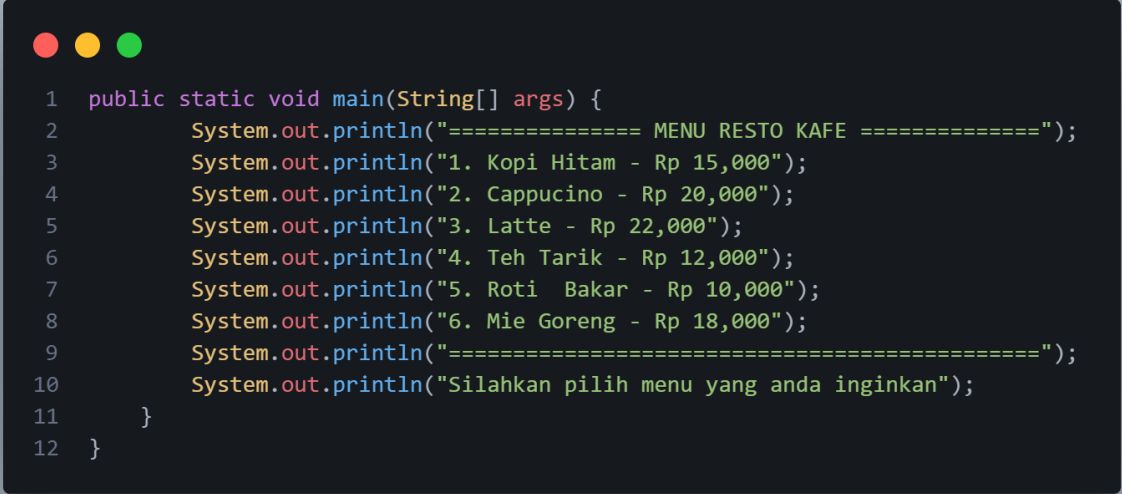
### **1. Apakah fungsi tanpa parameter selalu harus bertipe void?**

Tidak. Fungsi tanpa parameter bisa bertipe void (tidak mengembalikan nilai) *atau* bertipe lain jika ingin mengembalikan nilai. “Tanpa parameter” hanya berarti tidak ada argumen input, bukan soal tipe kembalian.

### **2. Apakah daftar menu pada program kafe dapat ditampilkan tanpa menggunakan fungsi Menu()?**

Tidak bisa

**Modifikasi kode program tersebut untuk dapat menampilkan daftar menu tanpa menggunakan fungsi!**



```
1  public static void main(String[] args) {  
2      System.out.println("===== MENU RESTO KAFE =====");  
3      System.out.println("1. Kopi Hitam - Rp 15,000");  
4      System.out.println("2. Cappuccino - Rp 20,000");  
5      System.out.println("3. Latte - Rp 22,000");  
6      System.out.println("4. Teh Tarik - Rp 12,000");  
7      System.out.println("5. Roti Bakar - Rp 10,000");  
8      System.out.println("6. Mie Goreng - Rp 18,000");  
9      System.out.println("=====");  
10     System.out.println("Silahkan pilih menu yang anda inginkan");  
11 }  
12 }
```

### **3. Jelaskan keuntungan menggunakan fungsi Menu() dibandingkan menulis semua perintah penampilan menu langsung di dalam fungsi main.**

- Modularitas: memisahkan tanggung jawab.
- Reuse: bisa dipanggil dari banyak tempat.
- Mudah diuji dan dibaca.
- Jika ingin ubah tampilan, cukup ubah di satu tempat.

**4. Uraikan secara singkat alur eksekusi program ketika fungsi Menu() dipanggil dari main (mulai dari program dijalankan sampai daftar menu tampil di layar)**

- `main` menemukan pemanggilan `Menu()`.
- Kontrol berpindah ke badan `Menu()`, perintah cetak dieksekusi.
- Setelah selesai, kontrol kembali ke `main` dan program selesai.

## **PERCOBAAN 2**

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

untuk mengirim input tipe data ke fungsi sehingga fungsi bisa berperilaku berbeda berdasarkan input itu.

**2. Jelaskan mengapa pada percobaan ini fungsi Menu() menggunakan parameter namaPelanggan dan isMember?**

Nama pelanggan menampilkan nama menggunakan `String` sedangkan `isMember` menampilkan nilai `true/false` yaitu `Boolean`

**3. Apakah parameter sama dengan variabel? Jelaskan.**

Parameter adalah variabel lokal pada fungsi yang menerima nilai dari pemanggil. Mereka *mirip* variabel tetapi cakupannya lokal hanya di fungsi itu.


**4. Jelaskan bagaimana cara kerja parameter isMember pada fungsi Menu(). Apa perbedaan output ketika isMember bernilai true dan ketika false?**

Jika `true` maka bagian kondisi member dijalankan ( menampilkan pesan diskon), jika `false` jalur lain dijalankan..

**5. Apa yang akan terjadi jika memanggil fungsi Menu() tanpa menyertakan parameter namaPelanggan dan isMember?**

Kompilator akan error karena metode dengan `Menu(String, boolean)` mengharuskan dua argumen. Anda perlu membuat overload tanpa parameter atau sesuai panggilan.

**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.**




```

1  public static void Menu(String namaPelanggan, String kodePromo, boolean isMember) {
2      System.out.println("Selamat datang, " + namaPelanggan + "!");
3      if (isMember) {
4          System.out.println("Anda adalah member");
5      }
6      if (kodePromo == "diskon50") {
7          System.out.println("dapatkan diskon 50% untuk setiap pembelian!");
8      } else if (kodePromo == "diskon30") {
9          System.out.println("dapatkan diskon 30% untuk setiap pembelian!");
10     } else {
11         System.out.println("Kode invalid");
12     }

```

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.



```

1  public static void main(String[] args) {
2      Menu("Fia", "diskon50", true);
3      System.out.println();
4      Menu("Budi", "diskon30", true);
5  }

```

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.

Ya — parameter memisahkan data dan logika, memudahkan pengujian dan pengembangan.

### PERCOBAAN 3

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.

Butuh bila fungsi menghasilkan sebuah nilai yang akan dipakai di tempat lain. Tidak perlu jika fungsi hanya melakukan aksi (mis. menampilkan menu) dan tidak perlu memberi nilai kembali — contoh pada :

```
1 int[] hargaItems = {15_000, 20_000, 22_000, 12_000, 10_000, 18_000};
2
3     int hargaTotal = hargaItems[pilihanMenu - 1] * banyakItem;
4     return hargaTotal;
```


```
1 public static void main(String[] args) {
```

2. Fungsi `hitungTotalHargaNoAbsen` 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.

Tipe data return dan parameter pada `hitungTotalHargaNoAbsen`:

- Tipe return: `double` (nilai total).
- Parameter contoh: `int pilihanMenu` (nomor menu yang dipilih), `int banyakPesanan` (jumlah porsi). Kedua parameter ini dipakai untuk menentukan harga dasar sebelum diskon.

3. Modifikasi kode di atas sehingga fungsi `hitungTotalHargaNoAbsen` 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`.



```
1  System.out.print("Masukkan Kode Diskon Anda :");
2      String kodePromo = sc.nextLine();
3      if (kodePromo.equalsIgnoreCase("diskon30")) {
4          diskon=0.3;
5      }else if (kodePromo.equalsIgnoreCase("diskon50")) {
6          diskon=0.5;
7      }else {
8          System.out.println("Kode invalid!!!");
9      }
```

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)?

```

1  System.out.print("Masukkan Kode Diskon Anda :");
2  String kodePromo = sc.nextLine();
3  if (kodePromo.equalsIgnoreCase("diskon30")) {
4      diskon = 0.3;
5  } else if (kodePromo.equalsIgnoreCase("diskon50")) {
6      diskon = 0.5;
7  } else {
8      System.out.println("Kode invalid!!!");
9  }
10 System.out.print("Berapa jenis menu yang ingin Anda pesan : ");
11 int jenis = sc.nextInt();
12
13 double totalKeseluruhan = 0;
14 for (int i = 1; i <= jenis; i++) {
15     System.out.println("Pesanan ke- " + i);
16
17     System.out.print("Masukkan nomor menu (1-6) : ");
18     int pilihMenu = sc.nextInt();
19     System.out.print("Masukkan jumlah item yang dipesan : ");
20     int jumlahItem = sc.nextInt();
21 }
22 double totalMenu = hitungTotalHarga04(pilihanMenu, banyakItem, 0);
23 totalKeseluruhan += totalMenu;
24 double setelahDiskon = totalKeseluruhan - (totalKeseluruhan * diskon);
25 System.out.println("\n=====");
26 System.out.println("TOTAL SEBELUM DISKON : Rp " + totalKeseluruhan);
27 System.out.println("DISKON : " + (diskon * 100) + "%");
28 System.out.println("TOTAL AKHIR : Rp " + setelahDiskon);
29 System.out.println("=====");
30

```

## PERCOBAAN 4

1. Jelaskan mengapa penulisan parameter di praktikum 4 ditulis dengan String... namaPengunjung!

karena perlakuan seperti array String[].

2. Modifikasi method daftarPengunjung menggunakan for-each loop.

```

1 public static void daftarPengunjung(String...namaPengunjung) {
2     System.out.println("Daftar Nama Pengunjung : ");
3     for (String a : namaPengunjung) {
4         System.out.println("- " + a);
5     }
6 }

```

3. Bisakah menggunakan dua tipe data varargs dalam satu fungsi? Jelaskan jawaban Anda berdasarkan aturan varargs di Java, dan berikan contohnya!

Tidak. Hanya boleh satu varargs parameter dan harus menjadi parameter terakhir.  
Contoh :

```

1 public static void daftarPengunjung(String...namaPengunjung) {
2 }
3 public static void daftarPengunjung(String...namaPengunjung,int...nomorPengunjung) {
4 }

```

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?

Tidak error tetapi tetap berjalan , dan output yang dihasilkan kosong.

## PERCOBAAN 5

1. Sebutkan tahapan dan urutan eksekusi program pada Percobaan 5, mulai dari program dijalankan hingga menampilkan luas persegi panjang dan volume balok!

Input panjang → input lebar → input tinggi → panggil fungsi hitung luas → panggil fungsi hitung volume → tampilkan hasil.

2. Tuliskan output yang dihasilkan program `HitungBalokNoAbsen` jika pengguna memasukkan panjang = 4, lebar = 3, dan tinggi = 5. Jelaskan secara singkat alur jalannya program sampai output tersebut muncul.

$\text{Luas} = 4 * 3 = 12$

$\text{Volume} = 4 * 3 * 5 = 60$

Luas persegi panjang = 12 & Volume balok = 60

3. Apakah output dari program di bawah ini kemudian jelaskan alur jalannya program tersebut!

Fungsi dengan parameter dan fungsi yang mengembalikan nilai sangat penting digunakan ketika sebuah proses membutuhkan data dari luar dan harus menghasilkan nilai yang bisa dipakai kembali. Pada percobaan ini, fungsi `hitungLuas(p, l)` dan `hitungVolume(p, l, t)` memakai parameter karena perhitungan luas dan volume membutuhkan nilai panjang, lebar, dan tinggi yang diberikan oleh pengguna. Kedua fungsi tersebut juga menggunakan **return value** karena hasil perhitungannya perlu dikembalikan ke `main` untuk ditampilkan. Dengan demikian, penggunaan parameter dan return value membuat program lebih fleksibel, jelas, dan terstruktur.

4. Pada saat apakah fungsi yang kita buat harus menggunakan parameter atau tidak? Pada saat apakah fungsi yang kita buat harus memiliki nilai kembalian atau tidak? Jelaskan!

`main()` tidak memiliki parameter panjang, lebar, dan tinggi karena formatnya sudah baku oleh Java, input diterima melalui Scanner, dan semua perhitungan dilakukan oleh fungsi khusus yang memang menerima parameter tersebut.

5. Jelaskan kapan sebuah fungsi sebaiknya menggunakan parameter dan kapan fungsi boleh tanpa parameter, dengan mengacu pada fungsi `hitungLuas` dan `hitungVolume` di Percobaan 5.

Tidak, hasil perhitungan luas dan volume *tidak akan berubah* meskipun urutan pemanggilan fungsi diubah, selama nilai input panjang, lebar, dan tinggi tetap sama.

6. Jelaskan kapan sebuah fungsi sebaiknya memiliki nilai kembalian (return value) dan kapan tidak perlu memiliki nilai kembalian, dengan mengacu pada fungsi `main`, `hitungLuas`, dan `hitungVolume`.



Dua fungsi diperlukan agar program lebih terstruktur, rumus tidak tercampur, mudah digunakan kembali, mudah diuji, dan mengikuti prinsip satu fungsi satu tugas.