

Nama : Faisal Rizky

Kelas : TI-1H

Laporan Jobsheet 11

Percobaan 1

1. Apakah fungsi tanpa parameter selalu harus bertipe void?

Jawaban:

Tidak

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!

Jawaban :

Bisa.

```
public class Kafe9 {  
    public static void main(String[] args) {  
  
        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("=====");  
        System.out.println("Silakan pilih menu yang Anda inginkan.");  
    }  
}
```

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

Jawaban :

- Saat ingin menampilkan daftar menu di bagian lain program (misalnya setelah user selesai memesan dan ingin memesan lagi), cukup memanggil Menu(); satu baris saja. tidak perlu menyalin-tempel (copy-paste) 10 baris kode print tersebut berulang kali.
 - Fungsi main menjadi lebih bersih dan ringkas.
 - Jika harga kopi berubah, hanya perlu mengeditnya di satu tempat (di dalam fungsi Menu)
4. Uraikan secara singkat alur eksekusi program ketika fungsi Menu() dipanggil dari main (mulai dari program dijalankan sampai daftar menu tampil di layar).

Jawaban :

- Program dimulai dari fungsi main.
- Java mengeksekusi baris kode di dalam main dan menemukan panggilan fungsi Menu();.
- Program melompat (berpindah) dari main masuk ke dalam definisi fungsi public static void Menu().

- Program menjalankan baris-baris kode di dalam fungsi Menu secara berurutan dari atas ke bawah (Mencetak "MENU RESTO KAFE", lalu "Kopi Hitam", dst sampai pesan "Silakan pilih...").
- Setelah baris terakhir di fungsi Menu selesai dieksekusi, program kembali ke fungsi main tepat di baris setelah panggilan Menu(); tadi.
- Karena tidak ada kode lagi di main, program selesai (terminated).

Percobaan 2

1. apakah kegunaan parameter di dalam fungsi?

Jawaban :

Parameter berguna sebagai input data dinamis untuk fungsi. Dengan parameter, sebuah fungsi dapat memproses data yang berbeda-beda setiap kali dipanggil tanpa harus mengubah kode di dalam fungsi tersebut. Ini membuat fungsi menjadi fleksibel.

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

Jawaban :

namaPelanggan: Digunakan agar program dapat menyapa pelanggan secara personal (misal: "Selamat datang, Andi!").

isMember: Digunakan untuk mengecek status keanggotaan. Jika true (member), program akan menampilkan info diskon khusus. Jika tidak ada parameter ini, program tidak bisa membedakan mana pelanggan biasa dan mana member.

3. Apakah parameter sama dengan variabel? Jelaskan.

Jawaban :

Ya, pada dasarnya parameter adalah variabel.

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

Jawaban :

Jika true: Blok kode di dalam if akan dijalankan, sehingga muncul teks: *"Anda adalah member, dapatkan diskon 10%..."*

Jika false: Blok kode di dalam if dilewati (tidak dijalankan), sehingga teks diskon tidak muncul.

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

Jawaban :

Akan terjadi Error (Compile Time Error). Jika fungsi didefinisikan membutuhkan 2 parameter (String dan boolean), maka saat pemanggilan wajib menyertakan 2 data tersebut. Jika dikosongkan Menu(), program tidak akan bisa dijalankan.

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.

Jawaban 6 & 7:

```
public class Kafe9 {
    public static void menu(String namaPelanggan, boolean isMember, String kodePromo) {
        System.out.println("Selamat datang, " + namaPelanggan + "!");

        if (isMember) {
            System.out.println(x: "Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");
        }

        System.out.println(x: "===== PENGECEKAN PROMO =====");
        if (kodePromo.equals(anObject: "DISKON50")) { Convert to switch
            System.out.println(x: "Berikan diskon 50%");
        } else if (kodePromo.equals(anObject: "DISKON30")) {
            System.out.println(x: "Berikan diskon 30%");
        } else {
            System.out.println(x: "kode invalid");
        }

        System.out.println(x: "=====MENU RESTO KAFE =====");
        System.out.println(x: "1. Kopi Hitam - Rp 15,000");
        System.out.println(x: "2. Cappuccino - Rp 20,000");
        System.out.println(x: "3. Latte - Rp 22,000");
        System.out.println(x: "4. Teh Tarik - Rp 12,000");
        System.out.println(x: "5. Roti Bakar - Rp 10,000");
        System.out.println(x: "6. Mie Goreng - Rp 18,000");
        System.out.println(x: "=====");
        System.out.println(x: "Silahkan pilih menu yang Anda Inginkan");
    }
}

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

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.

Jawaban :

Ya, sangat membantu. Tanpa parameter, harus membuat fungsi berbeda untuk setiap orang (misal: MenuBudi(), MenuAndi()) atau mengubah kode setiap kali ganti pelanggan

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.

Jawaban :

Butuh Return Value: Ketika hasil proses dari fungsi tersebut akan digunakan lagi untuk perhitungan atau logika lain di dalam program.

Contoh: **hitungTotalHarga()** mengembalikan angka (**int**) karena nilai total harganya harus diketahui oleh fungsi main untuk ditampilkan atau dijumlahkan dengan pesanan lain.

Tidak Butuh (Void): Ketika fungsi hanya bertugas melakukan aksi "di tempat" (seperti menampilkan teks ke layar) dan hasilnya tidak perlu diolah lagi.

Contoh: Menu() bertipe void karena hanya menampilkan daftar menu ke layar.

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.

Jawaban :

Tipe Data Nilai Kembalian: `int`. **`int pilihanMenu`** dan **`int banyakItem`**

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)?

Jawaban no 3 & 4 :

```

import java.util.Scanner;

public class Kafe9 {
    public static void menu(String namaPelanggan, boolean isMember) {
        System.out.println("Selamat datang, " + namaPelanggan + "!");

        if (isMember) {
            System.out.println(x: "Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");
        }

        System.out.println(x: "-----MENU RESTO KAFE -----");
        System.out.println(x: "1. Kopi Hitam - Rp 15,000");
        System.out.println(x: "2. Cappuccino - Rp 20,000");
        System.out.println(x: "3. Latte - Rp 22,000");
        System.out.println(x: "4. Teh Tarik - Rp 12,000");
        System.out.println(x: "5. Roti Bakar - Rp 10,000");
        System.out.println(x: "6. Mie Goreng - Rp 18,000");
        System.out.println(x: "=====");
        System.out.println(x: "Silahkan pilih menu yang Anda inginkan");
    }

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

        int hargaTotal = hargaItems[pilihanMenu - 1] * banyakItem;

        System.out.println(x: "===== PENGECEKAN PROMO =====");
        if (kodePromo.equals(anObject: "DISKON50")) {
            System.out.println(x: "Selamat! Anda mendapat diskon 50%");
            hargaTotal = hargaTotal / 2;
        } else if (kodePromo.equals(anObject: "DISKON30")) {
            System.out.println(x: "Selamat! Anda mendapat diskon 30%");
            hargaTotal = hargaTotal - (hargaTotal*30/100);
        } else if (!kodePromo.isEmpty()) {
            System.out.println(x: "kode invalid");
        }
        return hargaTotal;
    }
}

```

```

Run main | Debug main | Run | Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);    Resource leak: 'sc' is never closed
    menu(namaPelanggan: "Andi", isMember: true);

    int totalKeseluruhan = 0;
    String lanjut;

    do{
        System.out.print(s: "\nMasukkan 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();

        System.out.println(x: "Masukkan kode promo (tekan enter jika tidak ada:");
        String kodePromo = sc.nextLine();

        int hargaTotal = hitungTotalHarga(pilihanMenu, banyakItem, kodePromo);

        totalKeseluruhan += hargaTotal;

        System.out.print(s: "Apakah Anda ingin memesan menu lain? (y/t): ");
        lanjut = sc.nextLine();

    }while (lanjut.equalsIgnoreCase(anotherString: "y"));

    System.out.println(x: "\n=====");
    System.out.println("Total Keseluruhan pesanan Anda : Rp" + totalKeseluruhan);

}

```

Percobaan 4

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

Jawaban :

Penulisan String... disebut dengan **Varargs** (Variable Arguments). Tujuannya adalah agar method daftarPengunjung bersifat fleksibel dan dapat menerima **jumlah argumen yang tidak terbatas** (bisa 0, 1, 2, atau banyak data sekaligus) tanpa harus membuat array secara manual terlebih dahulu saat memanggilnya.

2. Modifikasi method daftarPengunjung menggunakan for-each loop.

Jawaban :

```
public class PengunjungCafe9 {
    static void daftarPengunjung(String...namaPengunjung) {
        System.out.println(x: "Daftar Nama Pengunjung:");
        for ( String nama : namaPengunjung) {
            System.out.println("- "+ nama);
        }
    }
}

Run main | Debug main | Run | Debug
public static void main(String[] args) {
    daftarPengunjung(...namaPengunjung: "Ali", "Budi", "Citra");
    daftarPengunjung(...namaPengunjung: "Andi");
    daftarPengunjung(...namaPengunjung: "Doni", "Eti", "Fahmi", "Galih");
}
}
```

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

Jawaban :

Tidak bisa

- Dalam satu method, hanya boleh ada **satu** parameter varargs.
- Parameter varargs harus selalu diletakkan di posisi **paling akhir** (parameter terakhir).

Jika mencoba membuat dua varargs, compiler Java akan bingung menentukan batas akhir argumen pertama dan awal argumen kedua (ambigu).

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?

Jawaban :

Program tetap berjalan (tidak error) baik saat kompilasi maupun saat dijalankan. Jika dipanggil daftar **Pengunjung()**; (kosong), maka Java akan menganggap parameter **namaPengunjung** sebagai sebuah array kosong dengan panjang (length) 0.

Output yang dihasilkan: Hanya akan mencetak judulnya saja, tanpa ada daftar nama di bawahnya, karena perulangan (loop) tidak akan dijalankan.

Tugas

1. Buatlah sebuah class KubusNoPresensi yang di dalamnya terdapat fungsi untuk menghitung volume kubus dan luar permukaan kubus! Commit dan push program Anda ke Github dengan pesan "Tugas1"

Jawaban :

```
1 public class Kubus9 {
2     public static int hitungVolume (int sisi) {
3         int hasilVolume = sisi * sisi * sisi;
4         return hasilVolume;
5     }
6     public static int hitungLuasPermukaan (int sisi) {
7         int hasilLuasPermukaan = 6 * sisi * sisi;
8         return hasilLuasPermukaan;
9     }
10    public static void cetakHasil () {
11        System.out.println(x: "==== Perhitungan Kubus ===");
12        System.out.println(x: "-----");
13    }
14
15    Run main | Debug main | Run | Debug
16    public static void main(String[] args) {
17        cetakHasil();
18
19        int sisi = 6;
20        int volumeKubus = hitungVolume(sisi);
21
22        int luasKubus = hitungLuasPermukaan(sisi);
23
24        System.out.println("Sisi Kubus      :" + sisi);
25        System.out.println("Volume Kubus   :" + volumeKubus);
26        System.out.println("Luas Permukaan Kubus :" + luasKubus);
27    }
28 }
```

2. Di suatu kelas, terdapat N mahasiswa yang mengikuti satu mata kuliah. Nilai tiap mahasiswa untuk mata kuliah tersebut akan diinputkan oleh pengguna dan disimpan dalam sebuah array 1 dimensi berukuran N. Buatlah sebuah class NilaiMahasiswaNoPresensi yang di dalamnya terdapat beberapa fungsi berikut:

a. Fungsi isianArray untuk:

- Mengisi elemen-elemen array dengan nilai yang diinputkan oleh user (jumlah elemen = N, dibaca dari input)

b. Fungsi tampilArray untuk:

- Menerima parameter berupa array nilai mahasiswa
- Menampilkan seluruh nilai yang telah diinputkan

c. Fungsi hitTot untuk:

- Menerima parameter berupa array nilai mahasiswa
- Menghitung dan mengembalikan total nilai seluruh mahasiswa (return tipe int atau double)

d. Fungsi main untuk:

- Membaca nilai N (jumlah mahasiswa) dari input pengguna
- Membuat array dengan ukuran N
- Memanggil fungsi isianArray, tampilArray, dan hitTot
- Menampilkan total nilai yang dikembalikan oleh fungsi hitTot Commit dan push program Anda ke Github dengan pesan "Tugas2"

Jawaban :

```
import java.util.Scanner;

public class NilaiMahasiswa9 {
    static Scanner sc = new Scanner(System.in);

    public static void isianArray (int [] kumpulanNilai) {
        System.out.println(x: "\n ----- INPUT DATA NILAI -----");

        for (int i=0; i < kumpulanNilai.length; i++) {
            System.out.println("Masukkan Nilai Mahasiswa ke-" + (i+1)+ ":");
            kumpulanNilai[i] = sc.nextInt();
        }
    }

    public static void tampilanArray (int[] kumpulanNilai) {
        System.out.println(x: "---- DAFTAR NILAI MAHASISWA ----");

        for (int i = 0; i < kumpulanNilai.length; i++) {
            System.out.println("Mahasiswa ke-" + (i+1) + ":" + kumpulanNilai[i]);
        }
    }

    public static int hitungTotal (int [] kumpulanNilai){
        int total = 0;

        for (int i = 0; i < kumpulanNilai.length; i++) {
            total = total + kumpulanNilai[i];
        }

        return total ;
    }

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        System.out.print(s: "Masukkan Jumlah Mahasiswa :");
        int jumlahMhs = sc.nextInt();

        int [] dataNilai = new int [jumlahMhs];

        isianArray(dataNilai);

        tampilanArray(dataNilai);

        int hasilTotal = hitungTotal(dataNilai);

        System.out.println("\nTotal Nilai Seluruh Mahasiswa:" + hasilTotal);
    }
}
```

3. Ibu Mariana mengelola sebuah kafe. Berikut adalah rekap penjualan 5 menu dari hari pertama hingga ketujuh:

	Hari ke 1	Hari ke 2	Hari ke 3	Hari ke 4	Hari ke 5	Hari ke 6	Hari ke 7
Kopi	20	20	25	20	10	60	10
Teh	30	80	40	10	15	20	25
Es Kelapa Muda	5	9	20	25	10	5	45
Roti Bakar	50	8	17	18	10	30	6
Gorengan	15	10	16	15	10	10	55

- a. Buatlah sebuah class RekapPenjualanCafeNoPresensi yang digunakan untuk menyimpan dan mengolah data penjualan 5 menu kafe dari hari pertama hingga hari ketujuh. Di dalam class tersebut, buat dan gunakan beberapa fungsi (method) berikut:
- Fungsi untuk menginputkan data penjualan setiap menu untuk setiap hari (petunjuk: menggunakan array 2 dimensi).
 - Fungsi untuk menampilkan seluruh data penjualan dari hari pertama hingga hari terakhir dalam bentuk tabel

- Fungsi untuk menampilkan menu yang memiliki total penjualan tertinggi selama tujuh hari (tampilkan nama menu dan total penjualannya).
- Fungsi untuk menampilkan rata-rata penjualan untuk setiap menu selama tujuh hari (tampilkan nama menu dan nilai rata-ratanya).
Gunakan fungsi main untuk:
 - Memanggil fungsi input data penjualan,
 - Menampilkan seluruh data penjualan,
 - Menampilkan menu dengan penjualan tertinggi, dan
 - Menampilkan rata-rata penjualan untuk setiap menu.

b. Modifikasi program pada 3 (a) sehingga jumlah hari penjualan dan jumlah menu dibaca dari input pengguna, lalu sesuaikan ukuran array dan proses perhitungan agar tetap berjalan dengan benar.

Jawaban a & b :

```

1  import java.util.Scanner;
2
3  public class RekapPenjualanKafe9 {
4      static Scanner sc = new Scanner(System.in);
5
6      public static void inputData (int [][] data, String [] namaMenu) {
7          System.out.println("==== INPUT DATA PENJUALAN =====");
8
9          for (int i = 0; i < data.length; i++) {
10             System.out.println("Input jumlah untuk menu: " + namaMenu[i] + ":");
11
12             for(int j = 0; j < data[i].length; j++) {
13                 System.out.print(" Hari ke-" + (j+1) + ":");
14                 data [i][j] = sc.nextInt();
15             }
16         }
17     }
18
19     public static void tampilanTabel (int [][] data, String [] namaMenu) {
20         System.out.println("\n==== REKAP DATA PENJUALAN =====");
21
22         System.out.print(s: "Menu\t\t");
23         for (int h = 1; h <= data[0].length; h++) {
24             System.out.print("H-" + h + "\t");
25         }
26         System.out.println();
27
28         for ( int i = 0; i < data.length; i++) {
29             System.out.print(namaMenu [i] + "\t\t");
30
31             for (int j = 0; j < data[i].length; j++) {
32                 System.out.print(data[i][j] + "\t");
33             }
34             System.out.println();
35         }
36     }
37
38     public static void menuTertinggi (int[][] data, String [] namaMenu) {
39         int maxPenjualan = 0;
40         String bestSeller = "";
41
42         for ( int i = 0; i < data.length; i++) {
43             int totalBaris = 0;
44
45             for ( int j = 0; j < data[i].length; j++) {
46                 totalBaris += data[i][j];
47             }
48
49             if ( totalBaris > maxPenjualan ) {
50                 maxPenjualan = totalBaris;
51                 bestSeller = namaMenu[i];
52             }
53         }
54     }
55

```

```

52     }
53 }
54 }
55
56 System.out.println(x: "\n>>> MENU BEST SELLER <<<");
57 System.out.println("Menu: " + bestSeller+ "(Total: " + maxPenjualan +)");
58 }
59
60
61 public static void hitungRataRata (int [][] data, String[] namaMenu) {
62     System.out.println(x: "\n=== RATA-RATA PENJUALAN ===");
63
64     for (int i = 0; i < data.length; i++) {
65         double total = 0;
66
67         for (int j = 0; j < data[i].length; j++) {
68             total += data[i][j];
69         }
70
71         double rata = total / 7;
72         System.out.printf(format: "%-15s : %.2f per hari\n",namaMenu[i],rata);
73     }
74 }
75
76
77 Run main | Debug main | Run | Debug
78 public static void main(String[] args) {
79     System.out.println(x: "----- SETUP KAFE -----");
80
81     System.out.print(s: "Masukkan Jumlah Menu: ");
82     int jumlahMenu = sc.nextInt();
83
84     System.out.print(s: "Masukkan Jumlah Hari: ");
85     int jumlahHari = sc.nextInt();
86
87     sc.nextLine();
88
89     String daftarMenu [] = new String[jumlahMenu];
90     int dataPenjualan [][] = new int [jumlahMenu] [jumlahHari];
91
92     System.out.println(x: "\n----- INPUT NAMA MENU -----");
93     for (int i = 0; i < jumlahMenu; i++){
94         System.out.print("Nama Menu ke- " + (i+1) + ":");
95         daftarMenu [i] = sc.nextLine();
96     }
97
98     inputData(dataPenjualan, daftarMenu);
99     tampilanTabel(dataPenjualan, daftarMenu);
100     menuTertinggi(dataPenjualan, daftarMenu);
101     hitungRataRata(dataPenjualan, daftarMenu);
102 }
103 }

```