

Jobsheet 11

Fungsi 1 (Pertemuan 14)



Nama : Salsabila Widyadhana

NIM : 254107020200

Kelas : TI-1H

Mata Kuliah : Praktikum Daspro

1. Percobaan 1: Membuat Fungsi Tanpa Parameter

The screenshot shows a Java IDE interface. In the code editor, there is a file named KAFE25.java containing the following code:

```
1 Public class KAFE25 {
2     public static void Menu() {
3         System.out.println("----- MENU RESTO KAFE -----");
4         System.out.println("1. Kopi Hitam\tRp 15,000");
5         System.out.println("2. Cappuccino\tRp 20,000");
6         System.out.println("3. Es Teh\tRp 18,000");
7         System.out.println("4. Teh Tarik\tRp 12,000");
8         System.out.println("5. Roti Bakar\tRp 10,000");
9         System.out.println("6. Mie Goreng\tRp 18,000");
10    System.out.println("7. Nasi Goreng\tRp 15,000");
11    System.out.println("-----");
12    System.out.println("Silakan pilih menu yang Anda inginkan.");
13 }
14
15 Run | Debug | Run main | Debug main
16 public static void main(String[] args) {
17     Menu(); // Memanggil fungsi Menu()
18 }
19 }
```

In the terminal window, the output of the program is displayed:

```
----- MENU RESTO KAFE -----
1. Kopi Hitam Rp 15,000
2. Cappuccino Rp 20,000
3. Es Teh Rp 18,000
4. Teh Tarik Rp 12,000
5. Roti Bakar Rp 10,000
6. Mie Goreng Rp 18,000
7. Nasi Goreng Rp 15,000
-----  
Silakan pilih menu yang Anda inginkan.  
PS C:\PraktikumDaspro>
```

Commit Github Percobaan 1 :



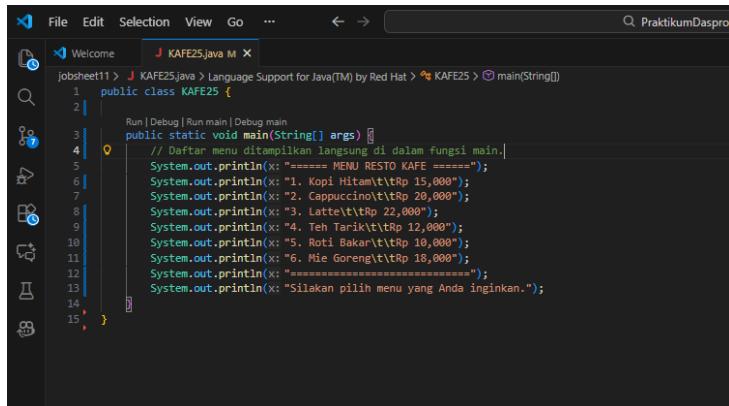
Pertanyaan :

1. Apakah fungsi tanpa parameter selalu harus bertipe void?
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
2. Ya, Daftar menu dapat ditampilkan tanpa menggunakan fungsi Menu(). Semua perintah penampilan menu (System.out.println(...)) dapat dituliskan langsung di dalam fungsi main.

Modifikasi Percobaan 1 :



```
File Edit Selection View Go ... PraktikumDaspro
Welcome J KAFE25.java M X
jobsheet11 > J KAFE25.java > Language Support for Java(TM) by Red Hat > KAFE25 > main(String[])
1 public class KAFE25 {
2
3     Run | Run main | Debug main
4     public static void main(String[] args) {
5         // Daftar menu ditampilkan langsung di dalam fungsi main.
6         System.out.println("===== MENU RESTO KAFE =====");
7         System.out.println("1. Kopi Hitam\tRp 15,000");
8         System.out.println("2. Cappuccino\tRp 20,000");
9         System.out.println("3. Latte\tRp 22,000");
10        System.out.println("4. Teh Tarik\tRp 12,000");
11        System.out.println("5. Roti Bakar\tRp 10,000");
12        System.out.println("6. Mie Goreng\tRp 18,000");
13        System.out.println("*****");
14        System.out.println("Silakan pilih menu yang Anda inginkan.");
15    }
}
PS C:\PraktikumDaspro> cd 'c:\PraktikumDaspro'; & '81c83debf2a40919707201f527e2d1f'rehat.java\jdt_ws\Prak
===== 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
*****
Silakan pilih menu yang Anda inginkan.
PS C:\PraktikumDaspro>

```

Commit Github Modifikasi Percobaan 1 :



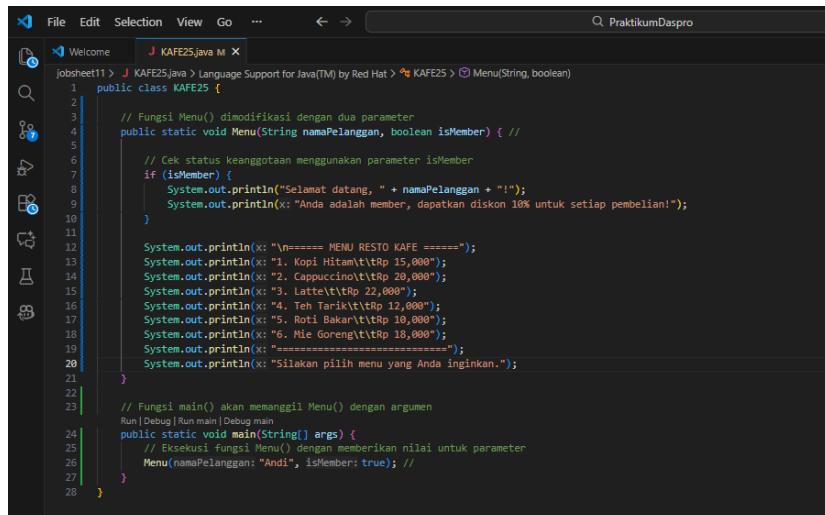
3. Keuntungan utama menggunakan fungsi `Menu()` adalah:

- Modularitas dan Keterbacaan: Fungsi `Menu()` mengelompokkan tugas spesifik (menampilkan menu) ke dalam blok kode yang terpisah, membuat fungsi `main` menjadi lebih ringkas dan mudah dipahami.
- Dapat Digunakan Kembali: Fungsi `Menu()` dapat dipanggil berkali-kali dari bagian program mana pun tanpa perlu menulis ulang baris-baris perintah menu yang sama.

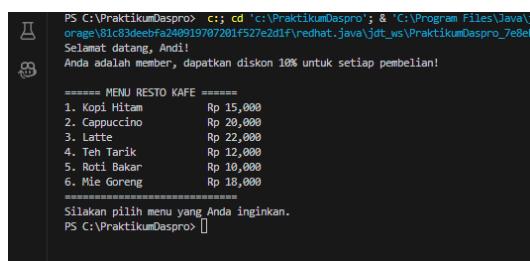
4. Alurnya adalah sebagai berikut :

- Mulai Program: Eksekusi program Java dimulai dari baris pertama fungsi `main (public static void main(String[] args))`.
- Panggilan Fungsi: Aliran kontrol mencapai baris `Menu();` di dalam `main`.
- Lompatan: Program menghentikan sementara eksekusi di `main` dan melompat ke baris pertama dari definisi fungsi `Menu()`.
- Eksekusi Fungsi: Semua perintah di dalam fungsi `Menu()` dieksekusi secara berurutan, menyebabkan daftar menu ditampilkan di layar.
- Kembali ke `main()`: Setelah semua baris di `Menu()` selesai dieksekusi, aliran kontrol kembali ke fungsi `main`, tepat setelah baris panggilan `Menu()`.
- Selesai: Program selesai dieksekusi karena tidak ada perintah lain di dalam `main`.

2. Percobaan 2: Membuat Fungsi Dengan Parameter



```
1 public class KAFE25 {
2
3     // Fungsi Menu() dimodifikasi dengan dua parameter
4     public static void Menu(String namaPelanggan, boolean isMember) { //
5
6         // Sek status keanggotaan menggunakan parameter isMember
7         if (isMember) {
8             System.out.println("Selamat datang, " + namaPelanggan + "!");
9             System.out.println("Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");
10        }
11
12        System.out.println("===== MENU RESTO KAFE =====");
13        System.out.println("1. Kopi Hitam\tRp 15,000");
14        System.out.println("2. Cappuccino\tRp 20,000");
15        System.out.println("3. Latte\tRp 22,000");
16        System.out.println("4. Teh Tarik\tRp 12,000");
17        System.out.println("5. Roti Bakar\tRp 10,000");
18        System.out.println("6. Mie Goreng\tRp 18,000");
19        System.out.println("===== ");
20        System.out.println("Silakan pilih menu yang Anda inginkan.");
21    }
22
23    // Fungsi main() akan memanggil Menu() dengan argumen
24    public static void main(String[] args) {
25        // Eksekusi fungsi Menu() dengan memberikan nilai untuk parameter
26        Menu("Andi", true); //
27    }
28 }
```



```
PS C:\PraktikumDaspro> cd "C:\PraktikumDaspro"; & 'C:\Program Files\Java\J
orange\81cb3dee\fa24991979707201f527e2d1f\redhat\java\jdt_ws\PraktikumDaspro_78eb
Selamat datang, Andi!
Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!

===== 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
=====
Silakan pilih menu yang Anda inginkan.
PS C:\PraktikumDaspro> [ ]
```

Commit Github Percobaan 2 :



Pertanyaan :

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.
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 di dalam fungsi adalah sebagai penampung nilai input yang dikirimkan ke fungsi saat fungsi tersebut dipanggil.
2. Fungsi `Menu()` menggunakan parameter `namaPelanggan (String)` dan `isMember (boolean)` karena fungsi tersebut perlu informasi dinamis tentang pelanggan untuk melakukan tugas tambahan selain menampilkan menu.
3. Parameter adalah jenis variabel khusus yang dibuat hanya untuk menerima data input ke fungsi, sementara variabel umum digunakan untuk penyimpanan dan pemrosesan data di dalam fungsi.

4. Cara Kerja `isMember`:

1. Ketika fungsi `Menu()` dipanggil, nilai `true` atau `false` dilewatkan ke parameter `isMember`.
2. Program mengecek kondisi `if (isMember)`.
3. Jika `isMember` bernilai `true`, blok kode di dalam `if` akan dieksekusi, yaitu menampilkan pesan diskon.
4. Jika `isMember` bernilai `false`, blok kode di dalam `if` dilewati, dan program langsung melanjutkan ke penampilan daftar menu.

Perbedaan Output:

- `isMember = true` (Contoh: `Menu ("Budi", true);`): Output akan menampilkan sapaan member dan pesan diskon 10% sebelum daftar menu.
- `isMember = false` (Contoh: `Menu ("Budi", false);`): Output hanya akan menampilkan sapaan biasa ("Selamat datang, Budi!") dan langsung diikuti oleh daftar menu, tanpa ada pesan diskon.

- 5.Ketika memanggil Menu() tanpa argumen, kompiler Java akan mendeteksi bahwa panggilan fungsi tersebut tidak cocok dengan tanda tangan fungsi yang didefinisikan, sehingga menghasilkan error yang menyatakan bahwa fungsi yang sesuai tidak dapat ditemukan (misalnya, *method Menu() in class KafeNoPresensi cannot be applied to given types; required: String, boolean; found: no arguments*).

6.Modifikasi Percobaan 2 beserta jawaban modifikasi kNo.7 :

```
1 public class KAFE25 {
2
3     public static void Menu(String namaPelanggan, boolean isMember, String kodePromo) {
4
5         System.out.println("Selamat datang, " + namaPelanggan + "!");
6
7         // Cek Status Member (isMember)
8         if (isMember) {
9             System.out.println("Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");
10        }
11
12        // LOSIKA DISKON KODE PROMO (Jawaban Pertanyaan No. 6)
13        if (kodePromo.equals("DISKON50")) {
14            // Tampilkan diskon 50%
15            System.out.println("Anda menggunakan kode promo DISKON50! Mendapatkan diskon 50%!");
16        } else if (kodePromo.equals("DISKON30")) {
17            // Tampilkan diskon 30%
18            System.out.println("Anda menggunakan kode promo DISKON30! Mendapatkan diskon 30%!");
19        } else if (kodePromo.isEmpty()) {
20            // Tampilkan kode invalid
21            System.out.println("Kode promo: " + kodePromo + " invalid atau tidak berlaku.");
22        }
23        // ----- AKHIR LOGIKA DISKON PROMO -----
24
25        System.out.println("----- MENU RESTO KAFE -----");
26        System.out.println("1. Kopi Hitam\tRp 15,000");
27        System.out.println("2. Cappuccino\tRp 20,000");
28        System.out.println("3. Latte\tRp 22,000");
29        System.out.println("4. Teh Tarik\tRp 12,000");
30        System.out.println("5. Roti Bakar\tRp 10,000");
31        System.out.println("6. Mie Goreng\tRp 18,000");
32
33        System.out.println("Silakan pilih menu yang Anda inginkan.");
34    }
35
36    // --- FUNGSI UTAMA (MAIN) ---
37    public static void main(String[] args) {
38
39        // Jawaban Pertanyaan No. 7
40        // Pelanggan "Budi", member (true), kode promo "DISKON30"
41        System.out.println("Panggilan Fungsi (Jawaban No. 7) ---");
42        Menu(namaPelanggan: "Budi", isMember: true, kodePromo: "DISKON30");
43
44        System.out.println("\n");
45
46        // Contoh Panggilan Tambahan: Candra, member, kode promo invalid ("HEMAT")
47        System.out.println("Contoh Panggilan Tambahan ---");
48        Menu(namaPelanggan: "Candra", isMember: true, kodePromo: "HEMAT");
49
50    }
51
52 }
53
54 }
```

```
PS C:\PraktikumDaspro> cd 'C:\PraktikumDaspro'; & 'C:\Program Files\Java\AppData\Roaming\Code\user\workspace\storage\b1c83deedfa240919707201f527e2df\'
--- Panggilan Fungsi (Jawaban No. 7) ---
=====
Selamat datang, Budi!
>>> Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!
>>> Anda menggunakan kode promo DISKON30! Mendapatkan diskon 30%!

===== 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
=====
Silakan pilih menu yang Anda inginkan.

--- Contoh Panggilan Tambahan ---
=====
Selamat datang, Candra!
>>> Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!
Kode promo: HEMAT invalid atau tidak berlaku.

===== 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
=====
Silakan pilih menu yang Anda inginkan.
PS C:\PraktikumDaspro>
```

8.Ya,

Alasan:

- Keterbacaan `main()`: Fungsi `main()` menjadi sangat jelas dan ringkas. Dengan melihat `Menu("Budi", true);`, pembaca kode langsung tahu bahwa Budi sedang disambut dan dia adalah anggota, tanpa perlu melihat detail internal fungsi `Menu()`.
- Fleksibilitas (Pengembangan): Jika ingin menyambut pelanggan baru (misalnya, `Menu("Citra", false);`), tidak perlu mengubah kode di dalam fungsi `Menu()`. Cukup memanggilnya dengan argumen yang berbeda. Jika nilai-nilai tersebut *hardcode* di dalam fungsi, harus mengubah dan mengompilasi ulang fungsi setiap kali pelanggan atau status keanggotaan berubah.

9.Commit Github Modifikasi Percobaan 2 :



3. Percobaan 3: Membuat Fungsi dengan Nilai Kembalian

```
File Edit Selection View Go ... < > x

Welcome > J_HitungTotalHarga25.java Language Support for Java(TM) by Red Hat > J_HitungTotalHarga25 > hitungTotalHarga(int, int)
jobabewati > J_HitungTotalHarga25.java [Language Support for Java(TM) by Red Hat] > J_HitungTotalHarga25 > hitungTotalHarga(int, int)
1 import java.util.Scanner;
2
3 public class HitungTotalHarga25 {
4
5     // Scanner untuk memasukkan input dari pengguna
6     static Scanner sc = new Scanner(System.in);
7
8     // --- Tampilan Pilihan Menu ---
9     public static void Main() {
10         System.out.println("----- MENU RESTO ABC -----");
11         System.out.println("1. Kopi Hitam@10k 10,000");
12         System.out.println("2. Cappuccino@15k 20,000");
13         System.out.println("3. Latte@15k 22,000");
14         System.out.println("4. Espresso@10k 15,000");
15         System.out.println("5. Roti Bakar@15k 10,000");
16         System.out.println("6. Mie Goreng@15k 18,000");
17         System.out.println("7. Es Teh@10k 10,000");
18     }
19
20     // --- PERCABIAN 1 (Dengan Nilai Kebulian) ---
21
22     // Fungsi untuk menghitung total harga pesanan.
23     // Parameter pertama: Nomor menu yang dipilih (1-6)
24     // Parameter kedua: Jumlah item yang dipesan
25     // Return: Total harga pesanan
26
27     public static int hitungTotalHarga(int pilihanMenu, int banyakItem) {
28         switch(pilihanMenu) {
29             case 1: return 10000 * banyakItem;
30             case 2: return 20000 * banyakItem;
31             case 3: return 22000 * banyakItem;
32             case 4: return 15000 * banyakItem;
33             case 5: return 10000 * banyakItem;
34             case 6: return 18000 * banyakItem;
35             case 7: return 10000 * banyakItem;
36             default: return -1; // Menghindari nilai total harga negatif
37         }
38     }
39
40     // Cek validitas pilihan menu
41     if (pilihanMenu >= 8 || pilihanMenu <= hargatotal.length) {
42         // Hitung harga total
43         int totalHarga = hargatotal[pilihanMenu - 1]; // -1 karena array dimulai dari indeks 0
44         int hargatalit = hargatotal + banyakItem;
45         return hargatalit; // Menghindari nilai total harga negatif
46     } else {
47         // Jika pilihan tidak valid
48         System.out.println("Pilihan menu tidak valid.");
49         return 0;
50     }
51
52     // --- FUNGSI UNTARA (MAIN) ---
53
54     Main();
55
56     // 1. Tampilkan Menu
57     // 2. Input dari pengguna
58     //    - Nomor menu dan jumlah item yang ingin Anda pesan (1-6); " ")
59     // 3. int pilihanMenu = sc.nextInt();
60
61     System.out.print("Masukkan jumlah item yang ingin dipesan: ");
62     int banyakItem = sc.nextInt();
63
64     // 4. Tampilkan Pesan
65     //    - Pesan yang menunjukkan jumlah nilai dan sistem hasilnya
66     int totalHarga = hitungTotalHarga(pilihanMenu, banyakItem);
67
68     // 5. Tampilkan hasil
69     if (totalHarga > 0) {
70         System.out.println("-----");
71         System.out.println("Total harga yang harus dibayar: Rp" + totalHarga);
72         System.out.println("-----");
73     }
74
75     sc.close();
76 }

```

```
PS C:\PraktikumDaspro> & 'C:\Program Files\Java/jdk-24\bin\javawspace-81833decfb2409919737201f527e2df.jar' redhat.java >jdt_w
===== 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
=====
Masukkan nomor menu yang ingin Anda pesan (1-6): 1
Masukkan jumlah item yang ingin dipesan: 2
-----
Total harga yang harus dibayar: Rp30000
-----
PS C:\PraktikumDaspro>
PS C:\PraktikumDaspro> [ ]
```

Commit Github Percobaan 3 :



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”

Jawab :

1. Membutuhkan Nilai Kembalian (Non-void): Ketika fungsi memiliki tugas untuk menghitung, memproses, atau mengambil suatu nilai, dan nilai hasil tersebut harus digunakan oleh bagian program lain yang memanggilnya.
 - Contoh dari Program Kafe: Fungsi `hitungTotalHarga(pilihanMenu, banyakItem)` harus mengembalikan nilai kembalian (tipe `int` atau `double`) karena hasil perhitungannya (total harga) diperlukan oleh fungsi `main` untuk ditampilkan kepada pengguna.
 2. - Parameter `1 int pilihanMenuNilai` input yang menunjukkan jenis menu mana yang dipesan oleh pelanggan (misalnya, 1 untuk Kopi Hitam, 3 untuk Latte). Nilai ini menentukan harga satuan.
 - Parameter `2 int banyakItemNilai` input yang menunjukkan jumlah atau kuantitas dari item menu yang dipilih. Nilai ini digunakan sebagai pengali dalam perhitungan total harga.
3. Modifikasi Percobaan 3 beserta jawaban modifikasi percobaan 3 no.4 :

```
PS C:\PraktikumDaspro> <: cd 'C:\PraktikumDaspro'; & 'C:\Program Files\Java\jdk-11.0.11\bin\java' -jar AppData\Roaming\Code\User\workspaceStorage\81c83deebf4a240919707281f5d4e0f\PROGRAM PEMESANAN KAFE (MULTI-MENU).jar
===== PROGRAM PEMESANAN KAFE (MULTI-MENU) =====

===== 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

Masukkan nomor menu yang ingin Anda pesan (1-6): 2
Masukkan jumlah item yang ingin dipesan: 11
Masukkan Kode Promo (kosongkan jika tidak ada): -
> Kode promo '-' invalid. Tidak ada diskon diterapkan.
> Subtotal Pesanan Ini: Rp220000
Apakah Anda ingin memesan menu lain? (y/t): y

===== 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

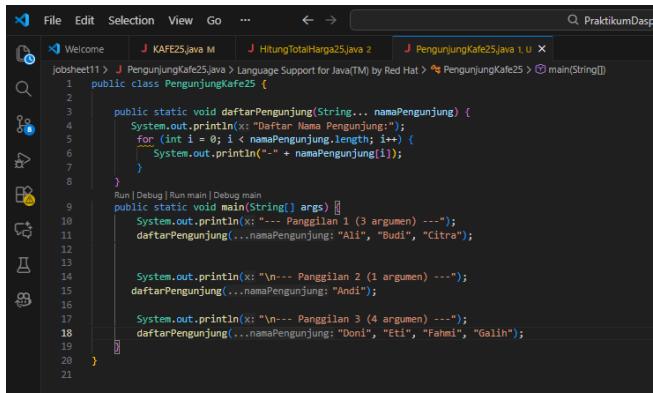
Masukkan nomor menu yang ingin Anda pesan (1-6): 1
Masukkan jumlah item yang ingin dipesan: 2
Masukkan Kode Promo (kosongkan jika tidak ada): DISKON30
> Kode Prom 'DISKON30' berhasil. Diskon: Rp9,000
> Subtotal Pesanan Ini: Rp210000
Apakah Anda ingin memesan menu lain? (y/t): t

===== TOTAL KESELURUHAN PESANAN ANDA: Rp241000 =====

PS C:\PraktikumDaspro>
```

5.

4. Percobaan 4: Fungsi Varargs

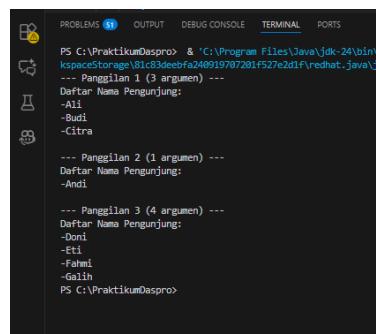


The screenshot shows an IDE interface with a code editor containing Java code. The code defines a class named PengunjungKafe25 with two methods: daftarPengunjung and main. The main method calls daftarPengunjung with three different argument lists: one with 3 arguments ('Ali', 'Budi', 'Citra'), one with 1 argument ('Andi'), and one with 4 arguments ('Doni', 'Eti', 'Fahmi', 'Galih'). The output of the program is displayed in the terminal below.

```
public class PengunjungKafe25 {
    public static void daftarPengunjung(String... namaPengunjung) {
        System.out.println("Daftar Nama Pengunjung:");
        for (int i = 0; i < namaPengunjung.length; i++) {
            System.out.println("- " + namaPengunjung[i]);
        }
    }
    public static void main(String[] args) {
        System.out.println("---- Panggilan 1 (3 argumen) ---");
        daftarPengunjung(...namaPengunjung: "Ali", "Budi", "Citra");

        System.out.println("---- Panggilan 2 (1 argumen) ---");
        daftarPengunjung(...namaPengunjung: "Andi");

        System.out.println("---- Panggilan 3 (4 argumen) ---");
        daftarPengunjung(...namaPengunjung: "Doni", "Eti", "Fahmi", "Galih");
    }
}
```



The terminal window shows the command PS C:\PraktikumDasp> & 'C:\Program Files\Java\jdk-24\bin\kspacedStorage\10c33eefaf24099197d7281f527e2d1f\redhat.java' being run. The output displays the names listed by the program for each call to daftarPengunjung.

```
PS C:\PraktikumDasp> & 'C:\Program Files\Java\jdk-24\bin\kspacedStorage\10c33eefaf24099197d7281f527e2d1f\redhat.java'
-- Panggilan 1 (3 argumen) ---
Daftar Nama Pengunjung:
-Ali
-Budi
-Citra

-- Panggilan 2 (1 argumen) ---
Daftar Nama Pengunjung:
-Andi

-- Panggilan 3 (4 argumen) ---
Daftar Nama Pengunjung:
-Doni
-Eti
-Fahmi
-Galih
PS C:\PraktikumDasp>
```

Commit Github Percobaan 4 :



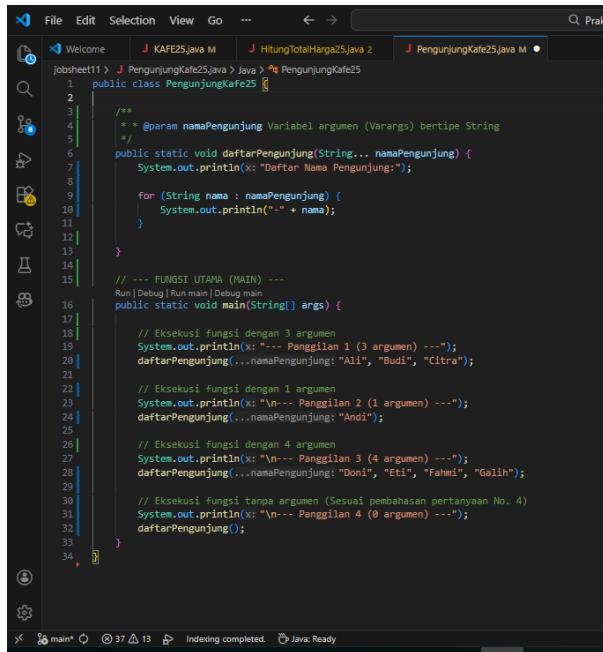
Pertanyaan :

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”

Jawab :

1. Java memperlakukan Varargs sebagai array satu dimensi dengan tipe data yang sesuai. Oleh karena itu, di dalam fungsi, namaPengunjung dapat diolah layaknya sebuah array,

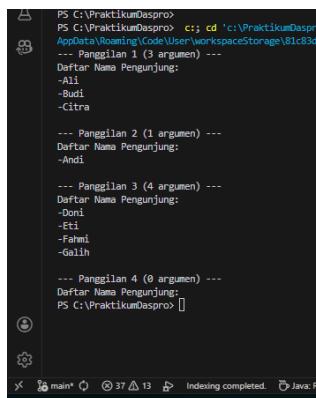
misalnya dengan menggunakan `namaPengunjung.length` untuk mengetahui jumlah argumen atau mengakses elemennya menggunakan indeks (`namaPengunjung[i]`).



The screenshot shows an IDE interface with several tabs open. The main code editor contains the following Java code:

```
1 public class PengunjungKafe25 {
2
3     /**
4      * @param namaPengunjung Variabel argumen (Varargs) bertipe String
5      */
6     public static void daftarPengunjung(String... namaPengunjung) {
7         System.out.println("Daftar Nama Pengunjung:");
8
9         for (String nama : namaPengunjung) {
10            System.out.println("-" + nama);
11        }
12    }
13
14    // --- FUNGSI UNTAHU (MAIN) ---
15
16    public static void main(String[] args) {
17
18        // Eksekusi fungsi dengan 3 argumen
19        System.out.println("... Panggilan 1 (3 argumen) ...");
20        daftarPengunjung(..namaPengunjung: "Ali", "Budi", "Citra");
21
22        // Eksekusi fungsi dengan 1 argumen
23        System.out.println("... Panggilan 2 (1 argumen) ...");
24        daftarPengunjung(..namaPengunjung: "Andi");
25
26        // Eksekusi fungsi dengan 4 argumen
27        System.out.println("... Panggilan 3 (4 argumen) ...");
28        daftarPengunjung(..namaPengunjung: "Doni", "Eti", "Fahmi", "Galih");
29
30        // Eksekusi fungsi tanpa argumen (Sesuai pembahasan pertanyaan No. 4)
31        System.out.println("... Panggilan 4 (0 argumen) ...");
32        daftarPengunjung();
33    }
34}
```

2.



The terminal window displays the output of the Java program:

```
PS C:\PraktikumDaspro> cd 'c:\PraktikumDaspro'
AppData\Roaming\Code\User\workspaceStorage\81c83d
--- Panggilan 1 (3 argumen) ---
Daftar Nama Pengunjung:
-Ali
-Budi
-Citra

--- Panggilan 2 (1 argumen) ---
Daftar Nama Pengunjung:
-Andi

--- Panggilan 3 (4 argumen) ---
Daftar Nama Pengunjung:
-Doni
-Eti
-Fahmi
-Galih

--- Panggilan 4 (0 argumen) ---
Daftar Nama Pengunjung:
PS C:\PraktikumDaspro> []
```

3. Tidak. Aturan Varargs di Java:

1. Sebuah fungsi (method) hanya boleh memiliki satu parameter varargs.
2. Jika parameter varargs digunakan, ia harus menjadi parameter terakhir dalam daftar parameter fungsi tersebut.

Alasan: Jika terdapat lebih dari satu varargs, atau jika varargs tidak diletakkan di akhir, kompiler tidak dapat menentukan di mana akhir dari argumen varargs pertama dan awal dari argumen varargs berikutnya. Ini akan menciptakan ambiguitas saat fungsi dipanggil.

1. Contoh yang salah menyebabkan error :

```
// ERROR: Tidak boleh ada dua varargs
```

```
public static void cetakData(String... nama, int... usia) { ... }
```

2. Contoh yang benar varargs harus diakhir :

// BENAR: Varargs diletakkan setelah parameter biasa

```
public static void cetakLaporan(String judul, String... dataIsian) {
```

```
// ...
```

```
}
```

4. Program akan tetap berjalan dan tidak akan mengalami *error* (baik saat kompilasi maupun saat dijalankan). Mekanisme Varargs: Parameter `String...` `namaPengunjung` secara otomatis akan diubah menjadi array kosong (`String[]`) dengan panjang nol (`namaPengunjung.length = 0`).

Output yang Dihasilkan:

- Program akan mencetak baris pertama: "Daftar Nama Pengunjung:".
- *Loop* (`for` atau `for-each`) untuk menampilkan nama tidak akan dijalankan sama sekali karena panjang array adalah nol.

5.



5. Tugas

1. Tugas 1

```
File Edit Selection View Go ...
Welcome J KAF25.java M J HitungTotalHarga25.java 2 J PengunjungKafe25.java J Kubus25.java 1 U ...
jobsheet11 > J Kubus25.java > Language Support for Java(TM) by Red Hat > Kubus25 > main(String[])
1 import java.util.Scanner;
2
3 public class Kubus25 {
4
5     /** Menghitung volume kubus
6      * @param sisi panjang sisi kubus (double)
7      * @return Volume kubus (double)
8      */
9     public static double hitungVolume(double sisi) {
10         // Rumus Volume (V) = sisi * sisi * sisi
11         return sisi * sisi * sisi;
12     }
13
14     /** Menghitung luas permukaan kubus
15      * @param sisi panjang sisi kubus (double)
16      * @return tipas permukaan kubus (double)
17      */
18     public static double hitungLuasPermukaan(double sisi) {
19         // Rumus Luas Permukaan (lp) = 6 * sisi * sisi
20         return 6 * sisi * sisi;
21     }
22
23     // --- FUNGSI UTAMA (MAIN) ---
24     Run | Debug | Run main | Debug main
25     public static void main(String[] args) {
26         Scanner sc = new Scanner(System.in);
27
28         System.out.println("----- PROGRAM HITUNG KUBUS -----");
29
30         // 1. Input panjang sisi dari pengguna
31         System.out.print("Masukkan panjang sisi kubus (cm): ");
32
33         // Menggunakan nextDouble agar bisa menerima input decimal
34         if (sc.hasNextDouble()) {
35             double sisi = sc.nextDouble();
36
37             if (sisi > 0) {
38                 // Memanggil fungsi hitungVolume
39                 double volume = hitungVolume(sisi);
40
41                 // * Memanggil fungsi hitungLuasPermukaan
42                 double luasPermukaan = hitungLuasPermukaan(sisi);
43
44                 // 4. Menampilkan hasil
45                 System.out.printf("Panjang Sisi : %.2f cm\n", sisi);
46                 System.out.printf("Volume Kubus : %.2f cm\u00b3\n", volume);
47                 System.out.printf("Luas Permukaan Kubus : %.2f cm\u00b2\n", luasPermukaan);
48
49             } else {
50                 System.out.println("Sisi kubus harus bernilai positif.");
51             }
52         } else {
53             System.out.println("Input tidak valid. Harap masukkan angka.");
54         }
55
56         sc.close();
57     }
58 }
```

```
PS C:\PraktikumDaspro> & 'C:\Program Files\Java\jdk-24\bin\fst2edifrednet.java' dt_ws\PraktikumDaspro_7e8ebba1bin
----- PROGRAM HITUNG KUBUS -----
Masukkan panjang sisi kubus (cm): 11
--- Hasil Perhitungan ---
Panjang Sisi : 11,00 cm
Volume Kubus : 1331,00 cm3
Luas Permukaan Kubus : 726,00 cm2
-----
PS C:\PraktikumDaspro>
```

Commit Github Tugas 1 :



2.Tugas 2

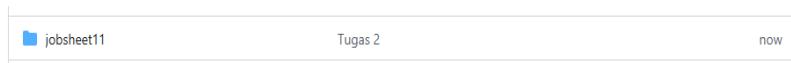
```
File Edit Selection View Go ... < >
Welcome J KAFES25.java M J HitungTotalKelas25.java J PrinsipPengGraf25.java J Kubus25.java I J HitungKelas25.java O
1 package dt_ws;
2
3 import java.util.Scanner;
4
5 public class NilaiMahasiswa {
6     static Scanner sc = new Scanner(System.in);
7
8     /** Mengisi elemen-elemen array dengan nilai yang diminta.
9      * @param nilaiArray Array 1 dimensi berisi double untuk nilai nilai.
10    */
11    public void isiNilai(double[] nilaiArray) {
12        System.out.print("~~~~~ Masukkan nilai Mahasiswa ~~~~~\n");
13        for (int i = 0; i < nilaiArray.length; i++) {
14            System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
15            nilaiArray[i] = sc.nextDouble();
16        }
17        System.out.println("~~~~~ Pengisian nilai selesai.\n");
18    }
19    /** Menulis parameter berupa array nilai mahasiswa dan menampilkan.
20     * @param nilaiArray Array 1 dimensi nilai mahasiswa.
21     */
22    public void cetakNilai(double[] nilaiArray) {
23        System.out.print("~~~~~ Cetak nilai Mahasiswa ~~~~~\n");
24        for (int i = 0; i < nilaiArray.length; i++) {
25            System.out.print("Mahasiswa ke-" + (i + 1) + ": " + nilaiArray[i]);
26        }
27        System.out.println("~~~~~\n");
28    }
29    /** Memberi parameter berupa array nilai mahasiswa, menghitung dan mengembalikan total nilaiarray.
30     * @param nilaiArray Array 1 dimensi nilai mahasiswa.
31     * @return Total nilai seluruh mahasiswa (double).
32     */
33    public static double hitungTotal(double[] nilaiArray) {
34        double totalNilai = 0;
35        // Loop untuk mengitung seluruh elemen array
36        for (double nilai : nilaiArray) {
37            totalNilai += nilai;
38        }
39        return totalNilai;
40    }
41
42    /* PADA SINI (MAIN) */
43    public static void main(String[] args) {
44        System.out.println("~~~~~ PROGRAM CLAH NILAI MAHASISWA ~~~~~");
45        // Meminta nilai N (Jumlah mahasiswa)
46        System.out.print("Masukkan jumlah Mahasiswa (N): ");
47        int N = sc.nextInt();
48        // Mahasiswa N positif
49        if (N < 0) {
50            System.out.println("Jumlah mahasiswa (N) harus lebih dari nol.");
51            return;
52        }
53        // Meminta array dengan ukuran N
54        nilai[] nilaiMahasiswa = new double[N];
55        // Mengambil fungsi isiNilaiArray
56        isiNilaiArray(nilaiMahasiswa);
57        // Mengambil fungsi cetakNilaiArray
58        cetakNilai(nilaiMahasiswa);
59        // Mengambil fungsi hitungTotal
60        double totalNilaiMahasiswa = hitungTotal(nilaiMahasiswa);
61
62        // Mengambil total nilai yang dikembalikan oleh fungsi hitungTotal
63        System.out.println("~~~~~ Hasil ~~~~~");
64        System.out.println("Total Nilai Keseluruhan Mahasiswa: " + totalNilaiMahasiswa);
65        System.out.println("~~~~~\n");
66    }
67
68    static void main(String[] args) {
69    }
70}
```

```
===== PROGRAM OLAH NILAI MAHASISWA =====
Masukan jumlah mahasiswa (N): 11

--- Memulai Pengisian Nilai Mahasiswa ---
Masukan nilai mahasiswa ke-1: 78
Masukan nilai mahasiswa ke-2: 88
Masukan nilai mahasiswa ke-3: 89
Masukan nilai mahasiswa ke-4: 76
Masukan nilai mahasiswa ke-5: 77
Masukan nilai mahasiswa ke-6: 67
Masukan nilai mahasiswa ke-7: 85
Masukan nilai mahasiswa ke-8: 83
Masukan nilai mahasiswa ke-9: 81
Masukan nilai mahasiswa ke-10: 76
Masukan nilai mahasiswa ke-11: 77
Pengisian nilai selesai.

--- Daftar Nilai Mahasiswa ---
Mahasiswa ke-1 : 78,00
Mahasiswa ke-2 : 88,00
Mahasiswa ke-3 : 89,00
Mahasiswa ke-4 : 76,00
Mahasiswa ke-5 : 77,00
Mahasiswa ke-6 : 67,00
Mahasiswa ke-7 : 85,00
Mahasiswa ke-8 : 83,00
Mahasiswa ke-9 : 81,00
Mahasiswa ke-10 : 76,00
Mahasiswa ke-11 : 77,00
-----
===== Total Nilai Keseluruhan Mahasiswa: 891,00 =====
PS C:\PraktikumDespros []
```

Commit Github Tugas 2 :



3.Tugas 3

```
File Edit Selection View Go ... PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
***** PROGRAM OLAH REKAP PENJUALAN CAFE *****
Masukkan jumlah menu yang akan diolah (N): 5
Masukkan jumlah hari penjualan: 1
... Masukkan Input Data Penjualan (modifikasi sb dengan input pengguna) ...
Masukkan nama menu ke-1: 2
Masukkan nama menu ke-2: 3
Masukkan nama menu ke-3: 4
Masukkan nama menu ke-4: 5
Masukkan nama menu ke-5: 6
... Masukkan Jumlah Penjualan per Hari ...
Menu: 2
Penjualan Hari ke-1: 2
Menu: 3
Penjualan Hari ke-1: 5
Menu: 4
Penjualan Hari ke-1: 11
Menu: 5
Penjualan Hari ke-1: 25
Menu: 6
Penjualan Hari ke-1: 7
Input data penjualan selesai.

***** REKAP PENJUALAN CAFE *****
-----|-----|-----|
MENU | HARI 1 | TOTAL |
-----|-----|-----|
2 | 2 | 2
3 | 5 | 5
4 | 11 | 11
5 | 25 | 25
6 | 7 | 7
-----|-----|-----|
... MENU DENGAN TOTAL PENJUALAN TERTINGGI ...
Menu Terlaris : 5
Total Penjualan : 25 item.

... RATA-RATA PENJUALAN HARIAN UNTUK SETiap MENU ...
2 : 2,00 item per hari
3 : 5,00 item per hari
4 : 11,00 item per hari
5 : 25,00 item per hari
6 : 7,00 item per hari
PS C:\PraktikumDaspro> [
```

Commit Github Tugas 3 :

