

Nama : Khania Puji Auliya

NIM : 254107020236

Prodi : D-IV Teknik Informatika

Kelas : 1G

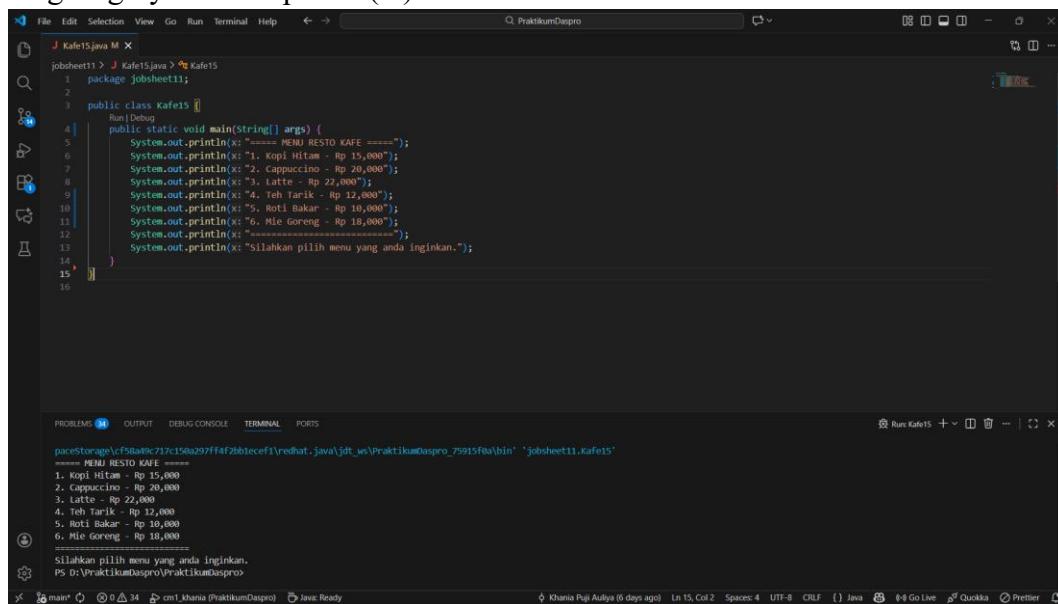
Percobaan 1

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, fungsi tanpa parameter bisa memiliki nilai kembalian.
2. Bisa. Fungsi hanyalah cara mengorganisir kode. Tanpa Menu() kita bisa menulis langsung System.out.println(...) di main.



```
File Edit Selection View Go Run Terminal Help < - > Q PraktikumDaspro
jobsheet11 > J Kafe15.java > Kafe15
1 package jobsheet11;
2
3 public class Kafe15 {
4     public static void main(String[] args) {
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. Nasi Goreng - Rp 18,000");
12        System.out.println("===== ");
13        System.out.println("Silahkan pilih menu yang anda inginkan.");
14    }
15 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
D:\PraktikumDaspro\jobsheet11>jdk-11.0.2\bin\java -jar "D:\PraktikumDaspro\jobsheet11\Kafe15.jar"
===== 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. Nasi Goreng - Rp 18,000
=====
silahkan pilih menu yang anda inginkan,
```

3. Organisasi & keterbacaan: main tetap ringkas; detail tampilannya berada di fungsi tersendiri.

Penggunaan ulang (reusability): fungsi menu() dapat dipanggil berkali-kali dari bagian program lain.

Pemeliharaan (maintainability): jika ingin mengubah daftar menu, cukup edit satu tempat.

Abstraksi: main fokus pada alur program, menu() fokus pada tampilan menu.

Testing & debugging lebih mudah: kamu bisa menguji fungsi menu() terpisah.

Memudahkan pengembangan fitur: mis. menambahkan parameter nama pelanggan, promo, atau format tampilan tidak mengotori main.

4. 1. Program Java dimulai dengan mengeksekusi statement pertama di dalam fungsi main().
2. Program dipanggil mencapai baris Menu(); di dalam main().
3. Program dieksekusi segera dialihkan ke baris pertama dari fungsi Menu().
4. Statement di dalam fungsi Menu() dieksekusi secara berurutan, yang terdiri dari statement System.out.println() yang mencetak daftar menu ke konsol.
5. Setelah statement terakhir di Menu() selesai dieksekusi (atau mencapai return), kontrol eksekusi dikembalikan ke main(), tepat setelah baris pemanggilanMenu().

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:

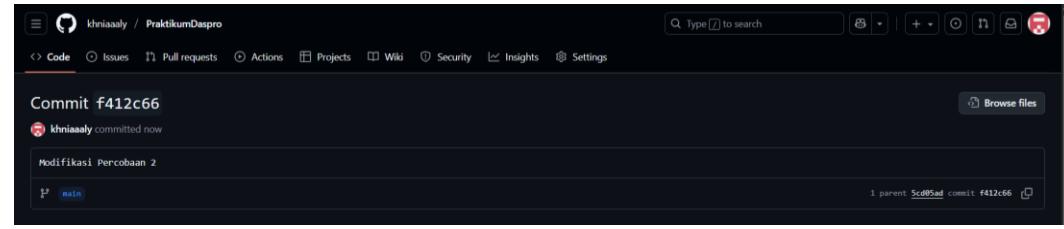
1. Parameter memungkinkan fungsi menerima input dari pemanggil sehingga fungsi menjadi generik dan fleksibel. Dengan parameter, satu fungsi bisa berperilaku berbeda bergantung nilai yang dikirim. Parameter menghindari hard-coding data di dalam fungsi, memperbaiki reusability dan modularitas.
2. namaPelanggan digunakan agar tampilan bisa personalisasi (mis. menampilkan "Selamat datang, Budi!"). Ini meningkatkan interaksi pengguna.
isMember digunakan untuk menentukan apakah menampilkan informasi diskon atau perlakukan khusus (mis. "Anda adalah member, dapat diskon 10%"). Dengan parameter ini fungsi tidak perlu mengecek sumber data global; cukup menerima status saat dipanggil.
3. Tidak sama, parameter didefinisikan pada deklarasi fungsi dan menjadi variabel lokal fungsi yang menerima nilai saat fungsi dipanggil. Sedangkan, variabel umum bisa dideklarasikan di banyak tempat: local variable, instance variable, atau static variable.
4. Saat Menu(nama, isMember) dipanggil, nilai boolean isMember dikopi ke parameter isMember di fungsi.
Di dalam fungsi, biasanya ada kondisi if (isMember) { ... } else { ... }.
Jika isMember == true: program mengeksekusi blok if — menampilkan teks terkait member, mis. "Anda adalah member, dapat diskon 10%".
Jika isMember == false: blok else dieksekusi atau tidak menampilkan pesan member, mis. menampilkan "Anda bukan member" atau tidak menampilkan info diskon.
5. Ketika memanggilnya tanpa argumen (Menu()), compiler tidak akan menemukan definisi fungsi yang cocok (yang tidak memiliki parameter) dan akan Error Kompilasi (compile-time error) pada program.

Fungsi Menu() saat didefinisikan wajib menerima dua parameter yaitu String dan boolean

```
File Edit Selection View Go Run Terminal Help <- > Q. PraktikumDispro
J Kafe15.java M
jobsheet11 > J Kafe15Java > Kafe15
1 package jobsheet11;
2
3 public class Kafe15 {
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         if (kodePromo.equalsIgnoreCase("DISKON50%")) {
12             System.out.println("Selamat! Anda mendapatkan DISKON 50%!");
13         } else if (kodePromo.equalsIgnoreCase("DISKON30%")) {
14             System.out.println("Selamat! Anda mendapatkan DISKON 30%!");
15         } else if (!kodePromo.isEmpty()) {
16             System.out.println("Kode promo invalid atau tidak berlaku.");
17         }
18     }
19
20     Run | Debug
21     public static void main(String[] args) {
22         Menu("Budi", true, "DISKON30");
23     }
24 }
```

- 6.
- 7.
8. Pada pengembangan fungsi Menu() dapat digunakan untuk siapapun ("Andi", "Budi") dan status member apapun (true or false) tanpa harus mengubah kodedidalam fungsi itu sendiri.

Sedangkan untuk kemudahan membaca: Saat memanggil Menu ("Budi", true, "DISKON30"); Kita langsung tahu konteksnya: Menu dipanggil untuk pelanggan "Budi" berstatus "member" dengan promo "DISKON30"



9. 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:

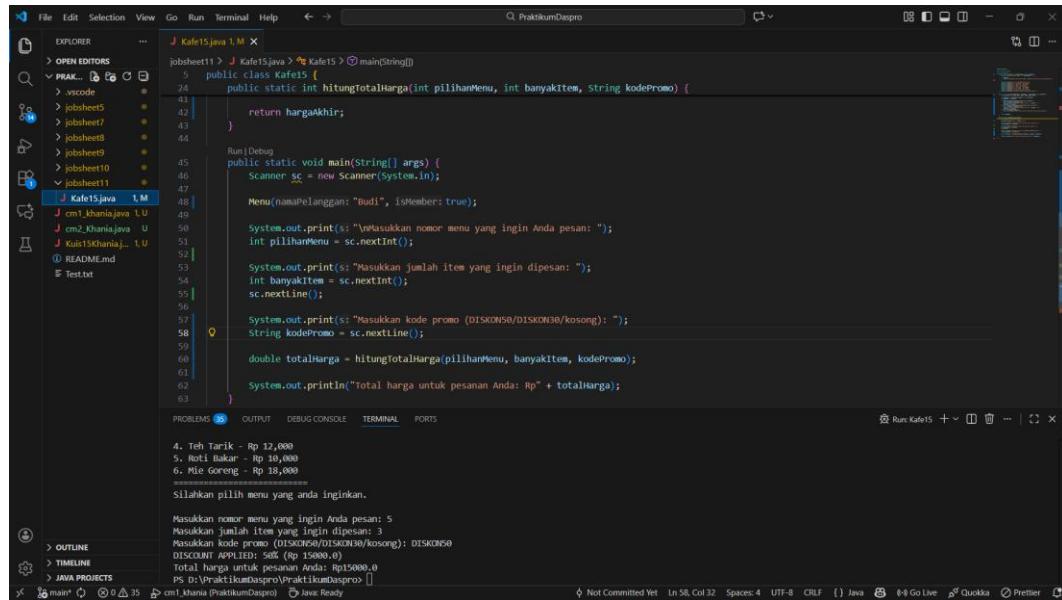
1. **Fungsi membutuhkan return value** ketika hasil perhitungan atau data yang dihasilkan diperlukan untuk langkah berikutnya. Contoh: hitungTotalHarga(pilihanMenu, banyakItem) harus mengembalikan int total agar main dapat menampilkan atau memprosesnya lebih lanjut (mis. menghitung kembalian, atau menerapkan diskon).
Fungsi tidak perlu return value (void) ketika tugas fungsi hanya melakukan aksi yang efeknya langsung terlihat (I/O) atau memodifikasi state internal tanpa perlu mengirim data kembali ke pemanggil. Contoh: tampilMenu() hanya menampilkan menu — tidak perlu return. Ringkas: kalau pemanggil butuh hasil → gunakan return; kalau hanya aksi/tampilan → void.

2. Tipe nilai kembalian: int (jumlah total harga dalam satuan rupiah). Bisa juga double jika ingin representasi pecahan (mis. bila ada pajak atau diskon menghasilkan pecahan).

Parameter:

- int pilihanMenu, indeks/nomor menu yang dipilih (1..n). Digunakan untuk memilih harga dari array harga.
- int banyakItem, jumlah porsi yang dipesan. Digunakan untuk mengalikan harga satuan agar memperoleh total untuk item tersebut.

Makna: fungsi mengalikan harga satuan (diperoleh dari pilihanMenu) dengan banyakItem lalu mengembalikan hasilnya.



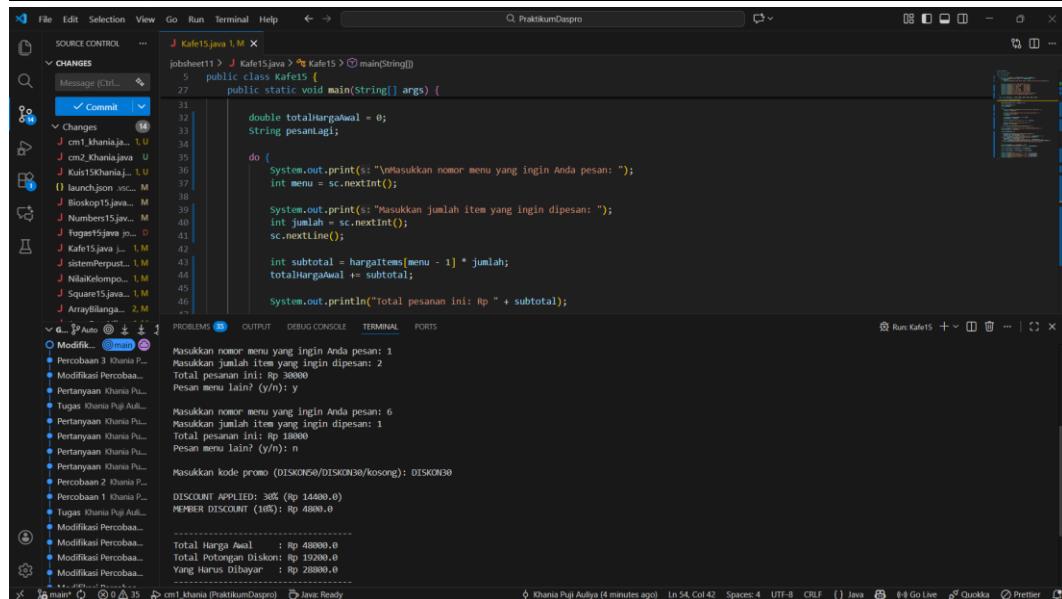
```

File Edit Selection View Go Run Terminal Help Q PraktikumDapro
EXPLORER > OPENEDATORS > PRAC... > vscode > jobsheet1 > J Kafe15.java 1, M
jobsheet11 > J Kafe15.java > Kafe15 > main(String[])
public class Kafe15 {
    public static int hitungTotalHarga(int pilihanMenu, int banyakItem, String kodePromo) {
        return hargaAkhir;
    }
}
Run|Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    Menu.namaPelanggan = "Budi", isMember = true;
    System.out.print("Masukkan nomor menu yang ingin anda pesan: ");
    int pilihanMenu = sc.nextInt();
    System.out.print("Masukkan jumlah item yang ingin dipesan: ");
    int banyakItem = sc.nextInt();
    sc.nextLine();
    System.out.print("Masukkan kode promo (DISKON50/DISKON30/kosong): ");
    String kodePromo = sc.nextLine();
    double totalHarga = hitungTotalHarga(pilihanMenu, banyakItem, kodePromo);
    System.out.println("Total harga untuk pesanan Anda: Rp" + totalHarga);
}

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
4. Telur Balik - Rp 12,000
5. Roti Bakar - Rp 5,000
6. Nasi Goreng - Rp 18,000
=====
Silahkan pilih menu yang anda inginkan.
Masukkan nomor menu yang ingin Anda pesan: 5
Masukkan jumlah item yang ingin dipesan: 2
Masukkan kode promo (DISKON50/DISKON30/kosong): DISKON50
DISCOUNT APPLIED: 50% (Rp 10000.0)
Total harga untuk pesanan Anda: Rp15000.0
PS D:\PraktikumDapro\PraktikumDapro] Java Ready

```

3.



```

File Edit Selection View Go Run Terminal Help Q PraktikumDapro
SOURCE CONTROL > CHANGES > Message (Ctrl+Shift+M)
Changes
J cm1_khana.java 1, 0
J cm2_khana.java 1, 0
J Kafe15.java 1, 0
J Kuis1Khana.java 1, 0
J launch.json 1, 0
J BlokOp15.java... M
J Numbers15.java... M
J Tugat15java.j... D
J Kafe15java.j... 1, M
J sistemPerpus... 1, M
J Nalikelopmo... 1, M
J Squre15java... 1, M
J ArrayBilang... 2, M
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Commit
Changes
Message (Ctrl+Shift+M)
J Kafe15.java 1, M
jobsheet11 > J Kafe15.java > Kafe15 > main(String[])
public class Kafe15 {
    public static void main(String[] args) {
        double totalHargaAwal = 0;
        String pesanLagi;
        do {
            System.out.print("Masukkan nomor menu yang ingin Anda pesan: ");
            int menu = sc.nextInt();
            System.out.print("Masukkan jumlah item yang ingin dipesan: ");
            int jumlah = sc.nextInt();
            sc.nextLine();
            int subtotal = hargaAtas[menu - 1] * jumlah;
            totalHargaAwal += subtotal;
            System.out.println("Total pesanan ini: Rp " + subtotal);
        } while (pesanLagi.equalsIgnoreCase("Y"));
        System.out.print("Masukkan nomor menu yang ingin Anda pesan: 1
Masukkan jumlah item yang ingin dipesan: 2
Total pesanan ini: Rp 30000
Pesan menu lain? (y/n): y
Masukkan nomor menu yang ingin Anda pesan: 6
Masukkan jumlah item yang ingin dipesan: 1
Total pesanan ini: Rp 18000
Pesan menu lain? (y/n): n
Masukkan kode promo (DISKON50/DISKON30/kosong): DISKON30
DISCOUNT APPLIED: 30% (Rp 14400.0)
MEMBER DISCOUNT (10%): Rp 4800.0
Total Harga Awal : Rp 48000.0
Total Potongan Diskon: Rp 19200.0
Yang Harus Dibayar : Rp 28800.0

```

4.

5.

Percobaan 4

Pertanyaan:

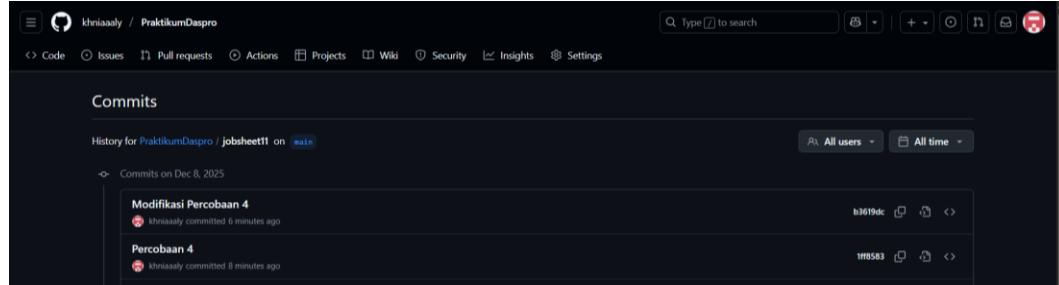
1. Jelaskan mengapa penulisan parameter di praktikum 4 ditulis dengan String... namaPengunjung!
2. Modifikasi method daftarPengunjung menggunakan for-each loop.
3. Bisakah menggunakan dua tipe data varargs dalam satu fungsi? Jelaskan jawaban Anda berdasarkan aturan varargs di Java, dan berikan contohnya!
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. String... namaPengunjung adalah sintaks varargs di Java yang memungkinkan pemanggilan fungsi dengan jumlah argumen variabel (0..n) dari tipe String. Varargs berguna untuk kasus ketika jumlah elemen tidak tetap.

2.

3. Tidak. Java hanya mengizinkan satu parameter varargs dan parameter varargs harus menjadi parameter terakhir dalam daftar parameter. Jadi tidak bisa method(int... a, String... b); hanya satu varargs yang diperbolehkan. Jika butuh dua koleksi berbeda, gunakan array atau collection untuk salah satu parameternya, mis. method(int[] a, String... b) atau method(String[] a, Integer[] b) tetapi hanya satu memakai
4. Program tetap berjalan. Varargs akan menjadi array kosong new String[0]. Tidak terjadi error pada kompilasi atau runtime. Output: jika implementasi menampilkan setiap nama dengan for-each, maka tidak ada baris yang ditampilkan. Jika ingin menampilkan pesan ketika tidak ada pengunjung, cek if (namaPengunjung.length == 0)



TUGAS

```

1.   public class Tugas1 {
2.
3.     public static int hitungVolume(int sisi) {
4.         return sisi * sisi * sisi;
5.     }
6.
7.     public static int hitungLuasPermukaan(int sisi) {
8.         return 6 * sisi * sisi;
9.     }
10.
11.    public static void main(String[] args) {
12.        Scanner sc = new Scanner(System.in);
13.
14.        System.out.print("Masukkan panjang sisi kubus: ");
15.        int sisi = sc.nextInt();
16.
17.        int volume = hitungVolume(sisi);
18.
19.        int luasPermukaan = hitungLuasPermukaan(sisi);
20.
21.        System.out.println("Volume kubus: " + volume + " cm");
22.        System.out.println("Luas permukaan kubus: " + luasPermukaan + " cm");
23.
24.    }
25.
26. }

```

PS D:\PraktikumDaspro\PraktikumDaspro> & 'C:\jdk-24.0.1\jdk-24.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Elvie Puji Fitriana\AppData\Roaming\CodeUser\workspaceStorage\cf58a9e71c15ba297ff4f2bb1ceef1\reshat.java\jdt_ws\PraktikumDaspro_75915fe0\bin' 'jobsheet11.Tugas1'
Masukkan panjang sisi kubus: 10
Volume kubus: 1000 cm
Luas permukaan kubus: 600 cm
PS D:\PraktikumDaspro\PraktikumDaspro>

```
File Edit Selection View Go Run Terminal Help < - > Q PraktikumDaspro
EXPLORER J Kafe15.java J PengunjungCafe5.java J Tugast.java J Tugas2.java
> OPEN EDITORS
PRAK... J Tugas2.java > Tugas2.java
> PRAK...
> vscode5
> jobsheet5
> jobsheet7
> jobsheet8
> jobsheet9
> jobsheet11
J Kafe15.java 1
J PengunjungCafe... 1
J Tugast.java 1
J Tugas2.java 1
cm1_khaniajava_1.U
cm2_khaniajava_U
Kuis150namej..._1.U
README.md
test.txt

public class Tugas2 {
    public static void isianArray(Scanner sc, int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print("Nilai mahasiswa ke-" + (i + 1) + ": ");
            arr[i] = sc.nextInt();
        }
    }

    public static void tampilArray(int[] arr) {
        System.out.println("== DAKTAR NILAI MAHASISWA ==");
        for (int i = 0; i < arr.length; i++) {
            System.out.println("Mahasiswa " + (i + 1) + ": " + arr[i]);
        }
    }

    public static int hitTot(int[] arr) {
        int total = 0;
        for (int v : arr)
            total += v;
        return total;
    }
}

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Nilai mahasiswa ke-10: 82
Nilai mahasiswa ke-11: 79
Nilai mahasiswa ke-12: 90

== DAKTAR NILAI MAHASISWA ==
Mahasiswa 1: 57
Mahasiswa 2: 90
Mahasiswa 3: 88
Mahasiswa 4: 78
Mahasiswa 5: 45
Mahasiswa 6: 96
Mahasiswa 7: 66
Mahasiswa 8: 65

```

2.

The screenshot shows a Microsoft Visual Studio Code interface with the following details:

- File Explorer:** Shows several Java files: `Kafe15.java`, `PengunjungCafe15.java`, `Tugas1.java`, `Tugas2.java`, and `Tugas3.java`. The file `Tugas3.java` is currently open.
- Code Editor:** Displays the `Tugas3.java` file which contains Java code for a supermarket application. The code includes methods for inputting sales data, displaying daily sales, and generating a receipt.
- Terminal:** Shows the command `cm1_khania@DESKTOP-1U: ~` and the output `Java Ready`.
- Status Bar:** Shows the current line (Ln 77), column (Col 1), and other status information like spaces, tabs, and file paths.

3.

