

PRAKTIKUM DASAR PEMROGRAMAN

“Laporan Pada Jobsheet 11”

Oleh:

Hafiz Rizqi Hernanda_1B

NIM (244107020154)



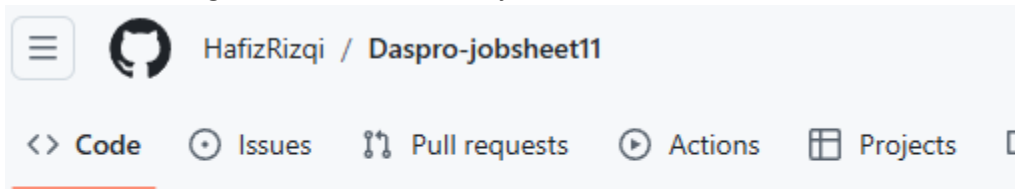
Jurusan Teknologi informasi

Teknik Informatika



Politeknik Negeri Malang

Percobaan 1

1. Buat repository baru dengan nama daspro-jobsheet11. Selanjutnya buat class baru, dan simpan file tersebut dengan nama KafeNoAbsen.java.



 **Daspro-jobsheet11** Public

 Kafe11.java >  Kafe11 >

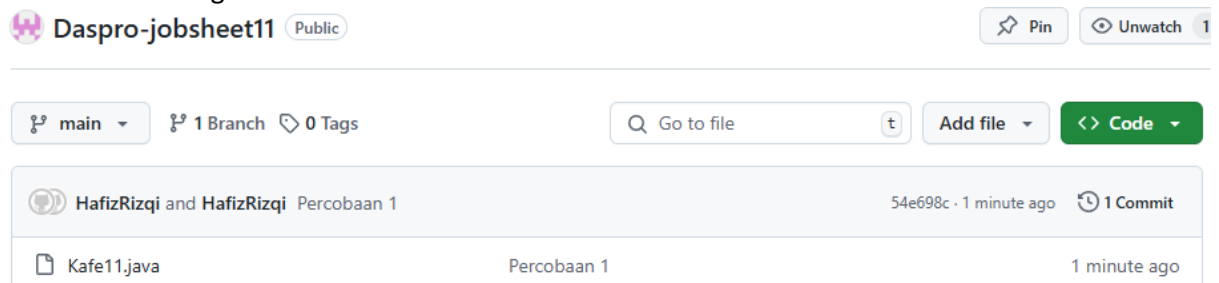
2. Buat fungsi Menu di dalam class tersebut.

```
public class Kafe11 {  
    public static void menu() {  
        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("Latte - Rp 22,000");  
        System.out.println("Teh Tarik - Rp 12,000");  
        System.out.println("Roti Bakar - Rp 10,000");  
        System.out.println("Mie Goreng - Rp 18,000");  
        System.out.println("=====");  
        System.out.println("Silahkan pilih menu yang anda inginkan");  
    }  
}
```

3. Buat fungsi main di dalam class tersebut, dan eksekusi fungsi Menu dari dalam fungsi main.

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

4. Push dan commit ke git



Jawaban Pertanyaan:

1. Apakah fungsi tanpa parameter selalu digunakan untuk fungsi "void" saja?

Tidak, **fungsi tanpa parameter** tidak selalu digunakan untuk fungsi yang bertipe **void**. Fungsi tanpa parameter adalah fungsi yang tidak menerima nilai masukan atau inputan dari luar, tetapi bisa saja mengembalikan nilai tertentu atau tidak mengembalikan apa pun (void).

2. Apakah bisa perintah menampilkan menu yang ada dituliskan tanpa fungsi Menu? modifikasi kode program tersebut untuk dapat menampilkan daftar menu tanpa menggunakan fungsi! Bisa, tinggal memasukkan datanya langsung di fungsi main.

```
public class Kafe11 {
    Run main | Debug main
    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("Latte - Rp 22,000");
        System.out.println("Teh Tarik - Rp 12,000");
        System.out.println("Roti Bakar - Rp 10,000");
        System.out.println("Mie Goreng - Rp 18,000");
        System.out.println("=====");
        System.out.println("Silahkan pilih menu yang anda inginkan");
    }
}
```

3. Apakah keuntungan menggunakan fungsi di dalam program?
Melakukan pemrograman akan lebih efisien, fleksibel dan mudah di kelola pada mengoperasikannya, dan dapat memilah mana yang terkena debug ataupun error

Percobaan 2

1. Ubah fungsi Menu dengan dua buah parameter bertipe String dan boolean di dalam class KafeNoAbsen.java.

```
J Kafe11.java > Kafe11 > main(String[] args)
1 public class Kafe11 {
2     public static void menu(String namaPelanggan, boolean isMember) {
3         System.err.println("Selamat datang. " + namaPelanggan + "! ");
4
5         if (isMember) {
6             System.err.println("Anda adalah Member, dapatkan diskon 10% untuk setiap pembelian!");
7         }
8         System.out.println("==== MENU RESTO KAFE ===");
9         System.out.println("1. Kopi Hitam - Rp 15,000");
10        System.out.println("2. Cappuccino - Rp 20,000");
11        System.out.println("Latte - Rp 22,000");
12        System.out.println("Teh Tarik - Rp 12,000");
13        System.out.println("Roti Bakar - Rp 10,000");
14        System.out.println("Mie Goreng - Rp 18,000");
15        System.out.println("=====");
16        System.out.println("Silahkan pilih menu yang anda inginkan");
17    }
}
```

2. Eksekusi fungsi `UcapanTambah` dari dalam fungsi `main`.

```
Run main | Debug main
public static void main(String[] args) {
    menu("Andi", true);
}
```

Jawaban Pertanyaan:

1. Apakah kegunaan parameter di dalam fungsi?
Parameter dalam fungsi memiliki kegunaan utama untuk mengirimkan data atau nilai dari inputan yang kita kasih dari fungsi `main` ke fungsi itu sendiri. Dengan adanya parameter, fungsi menjadi lebih fleksibel dan dapat beroperasi dengan data yang berbeda setiap kali dipanggil.
2. Apakah parameter sama dengan variabel? jelaskan!
Berbeda, parameter adalah jenis variabel yang khusus digunakan dalam fungsi/method untuk menerima data saat fungsi dipanggil. Meskipun parameter dan variabel memiliki kemiripan, fungsinya berbeda dalam konteks program. Parameter berfokus pada komunikasi antara fungsi, sedangkan variabel digunakan untuk berbagai keperluan logis dalam program.
3. Bagaimana cara kerja parameter `isMember` pada method `Menu`
Parameter `isMember` pada method `menu` berfungsi untuk menentukan apakah pelanggan yang memanggil method tersebut adalah seorang anggota (member) atau bukan.
Pada kondisi fungsi `main` untuk deklarasi `menu` dengan variabel boolean = `true`, yang berarti bahwa pelanggan tersebut merupakan member yang berhak mendapatkan diskon 10%.
4. Apa yang akan terjadi jika memanggil `Menu` tanpa menyertakan parameter `namaPelanggan` dan `isMember`?
program akan menghasilkan error saat kompilasi karena method `menu` yang Anda definisikan membutuhkan dua parameter (`namaPelanggan` dan `isMember`). Java mewajibkan jumlah, tipe, dan urutan parameter yang diberikan dalam pemanggilan method harus sesuai dengan deklarasi method.
5. 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

```
public static void main(String[] args) {
    menu();
}
```

```
if (kodePromo.equals("DISKON50")) {
    System.out.println("Kode promo valid: Anda mendapatkan diskon 50% untuk pembelian!");
} else if (kodePromo.equals("DISKON30")) {
    System.out.println("Kode promo valid: Anda mendapatkan diskon 30% untuk pembelian!");
} else {
    System.out.println("Kode promo tidak valid.");
}
```

```
Run main | Debug main
public static void main(String[] args) {
    menu("Andi", true, "DISKON50");
    System.out.println("");
    menu("Khesa", false, "DISKON30");
    System.out.println("");
    menu("Angga", false, "SALAHKODE");
    System.out.println("");
}
```

The screenshot shows the Visual Studio Code interface with the following components:

- Source Control:** Shows a commit message "Message (Ctrl+Enter to commit on...)" and a list of changes for "Kafe11.java".
- Output/Debug Console:** Displays the program's output. It shows three menu displays for different users: Andi (50% discount), Khesa (30% discount), and Angga (invalid code). The menu items and prices are: 1. Kopi Hitam - Rp 15,000, 2. Cappuccino - Rp 20,000, Latte - Rp 22,000, Teh Tarik - Rp 12,000, Roti Bakar - Rp 10,000, Mie Goreng - Rp 18,000.
- Terminal:** Shows the command prompt and the execution of the Java program.

Percobaan 3

1. Pada Class yang sama buat fungsi hitungTotalHargaNoAbsen di dalam class tersebut yang mengembalikan nilai total harga dan parameter masukan pilihan Menu dan banyaknya pesanan

```
hitungTotalHarga11.java > hitungTotalHarga11 > hitungTotalHarga11(int pilihanMenu, int banyakItem)
1 import java.util.Scanner;
2 public class hitungTotalHarga11 {
3     public static int hitungTotalHarga11 (int pilihanMenu, int banyakItem) {
4         int[] hargaItems = {15000, 20000, 22000, 12000, 10000, 18000};
5
6         int hargaTotal = hargaItems[pilihanMenu - 1] * banyakItem;
7         return hargaTotal;
8     }
```

2. Modifikasi fungsi main di dalam class tersebut, dan eksekusi fungsi `PenerimaUcapan` dari dalam fungsi main.

```
Run main | Debug main
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("\nMasukkan nomor menu yang ingin anda pesan");
    int pilihanMenu = sc.nextInt();
    System.out.print("Masukkan jumlah item yang ingin dipesan");
    int banyakItem = sc.nextInt();

    int totalHarga = hitungTotalHarga11(pilihanMenu, banyakItem);
    System.out.println("Total harga untuk pesanan anda: " + totalHarga);
}
```

Jawaban Pertanyaan:

1. Jelaskan mengenai, kapan suatu fungsi membutuhkan nilai kembalian (return)!
Fungsi membutuhkan **nilai kembalian** (return) ketika hasil dari proses di dalam fungsi tersebut perlu digunakan di luar fungsi, baik untuk ditampilkan, dihitung lebih lanjut, atau digunakan dalam logika program.
2. Modifikasi kode diatas sehingga fungsi `hitungTotalHarga` dapat menerima `kodePromo`. Jika `kodePromo` adalah "DISKON50", maka mendapat diskon 50% dari `totalHarga`. Jika `kodePromo` adalah "DISKON30", maka mendapat diskon 30% dari `totalHarga` tampilkan berikan diskon 30%. Jika tidak ada kode promo yang berlaku, tampilkan kode invalid dan tidak ada pengurangan total harga `totalHarga`.

```
if (kodePromo.equals("DISKON50")) {
    hargaTotal = hargaTotal / 2;
    System.out.println("Kode promo valid: Anda mendapatkan diskon 50%!");
} else if (kodePromo.equals("DISKON30")) {
    hargaTotal = hargaTotal - (hargaTotal * 30 / 100);
    System.out.println("Kode promo valid: Anda mendapatkan diskon 30%!");
} else {
    System.out.println("Kode promo tidak valid. Tidak ada diskon yang diberikan.");
}
```

3. Modifikasi kode diatas sehingga bisa memilih beberapa jenis menu berbeda serta menampilkan total keseluruhan pesanan.

```

Run main | Debug main
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int totalKeseluruhan = 0;
    String lanjutkan = "ya";

    while (lanjutkan.equals("ya")) {
        System.out.print("\nMasukkan nomor menu yang ingin Anda pesan (1-6): ");
        int pilihanMenu = sc.nextInt();
        System.out.print("Masukkan jumlah item yang ingin dipesan: ");
        int banyakItem = sc.nextInt();

        System.out.print("Masukkan kode promo (jika ada): ");
        sc.nextLine();
        String kodePromo = sc.nextLine();

        int totalHarga = hitungTotalHarga11(pilihanMenu, banyakItem, kodePromo);
        totalKeseluruhan += totalHarga;

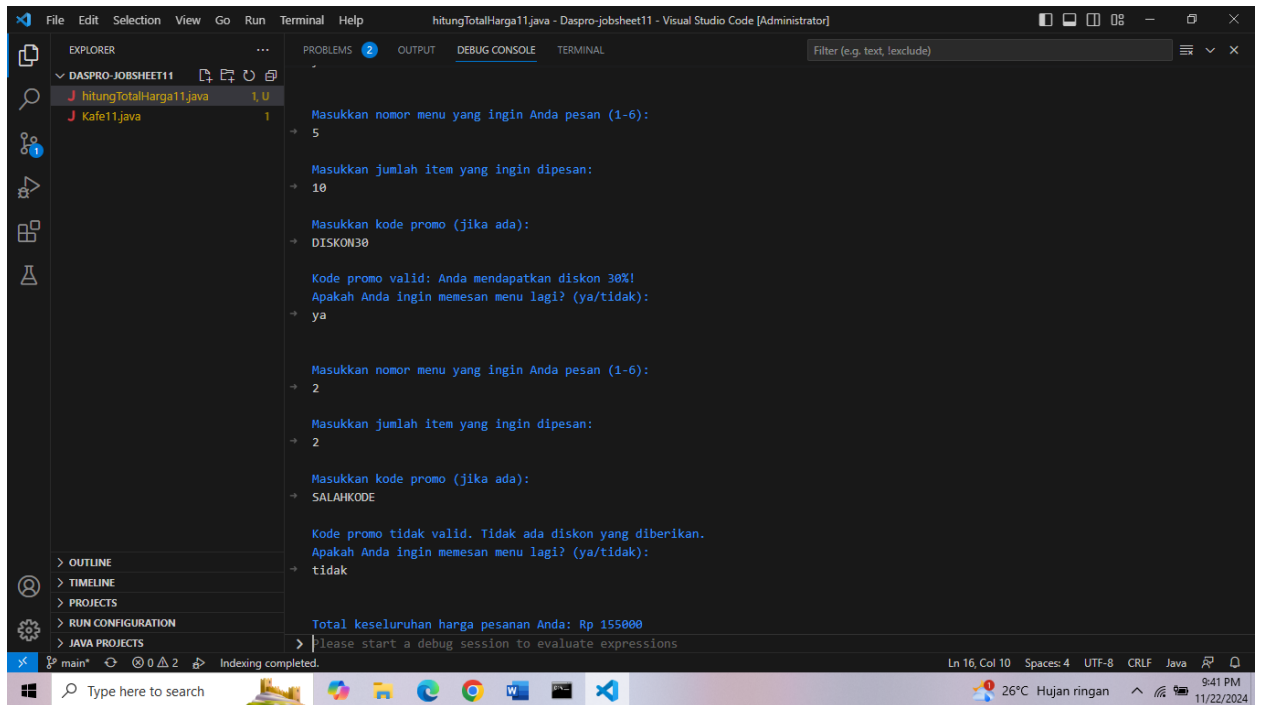
        System.out.print("Apakah Anda ingin memesan menu lagi? (ya/tidak): ");
        lanjutkan = sc.nextLine();
    }

    System.out.println("\nTotal keseluruhan harga pesanan Anda: Rp " + totalKeseluruhan);
}

```

Berikut hasil run untuk nomor 2 dan 3

The screenshot displays the Visual Studio Code interface during a Java program execution. The Explorer pane on the left shows the project structure with files 'hitungTotalHarga11.java' and 'Kafe11.java'. The Debug Console pane on the right shows the program's output, including prompts for menu number, item quantity, and promo code, followed by calculations and a final total price. The status bar at the bottom indicates the file is at line 16, column 10, with 4 spaces, UTF-8 encoding, and CRLF line endings.



Percobaan 4

1. Buat class baru, simpan file tersebut dengan nama `PengunjungCafeNoAbsen.java`

J PengunjungCafe11.java

2. Buatlah fungsi `daftarPengunjung` (bertipe void) di dalam class tersebut dengan menggunakan parameter `varArgs`

```

public class PengunjungCafe11 {
    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]);
        }
    }
}

```

3. Buat fungsi `main` di dalam class tersebut, dan eksekusi fungsi `namaPengunjung` dari dalam fungsi `main`.

```

Run main | Debug main
public static void main(String[] args) {
    daftarPengunjung("Ali", "Budi", "Citra");
}

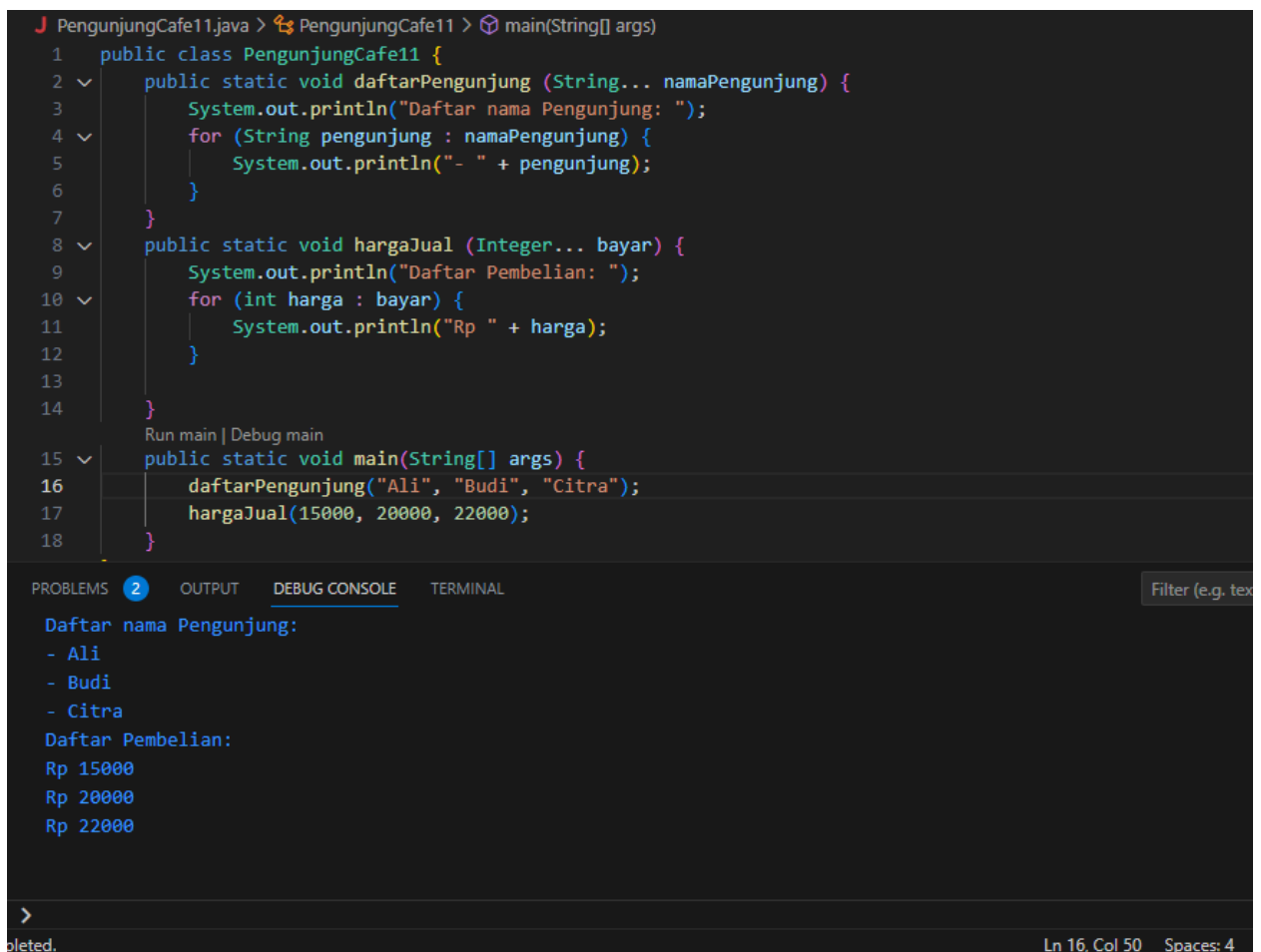
```


Jawaban Pertanyaan:

1. Jelaskan mengapa penulisan parameter di praktikum 4 di tulis dengan String... namaPengunjung!
Penulisan parameter fungsi dengan String... namaPengunjung memungkinkan fungsi untuk menerima jumlah argumen String yang tidak terbatas, memberi fleksibilitas lebih besar dalam pemanggilan fungsi. Ini menghindari penggunaan overload fungsi dan memungkinkan penulisan kode yang lebih ringkas dan mudah dikelola.
2. Modifikasi method daftarPengunjung menggunakan for-each loop!

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

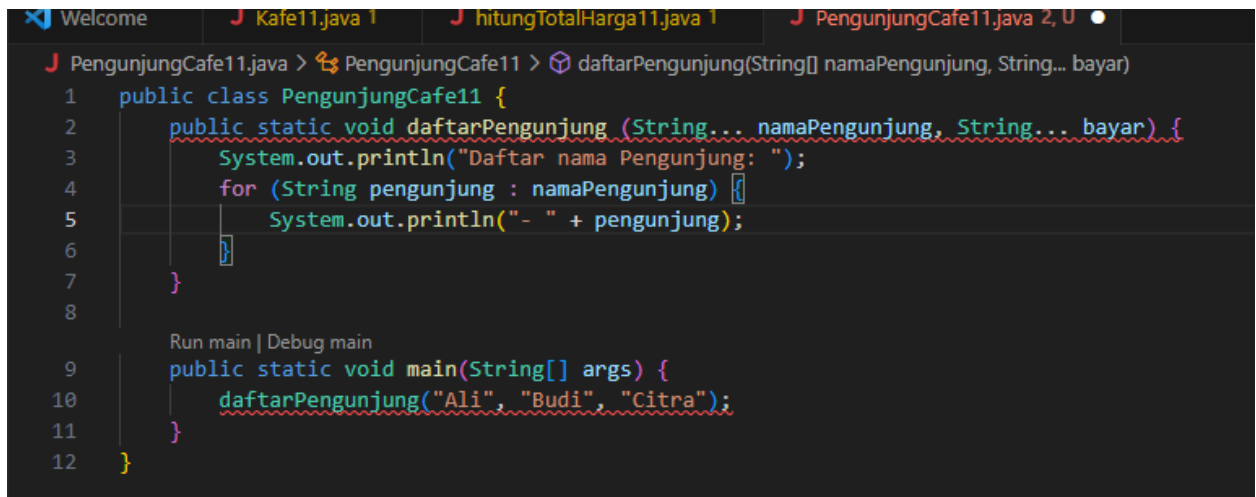
3. Bisakah menggunakan dua tipe data varargs dalam satu fungsi? Berikan contohnya!
Tidak bisa



```
J PengunjungCafe11.java > PengunjungCafe11 > main(String[] args)  
1 public class PengunjungCafe11 {  
2     public static void daftarPengunjung (String... namaPengunjung) {  
3         System.out.println("Daftar nama Pengunjung: ");  
4         for (String pengunjung : namaPengunjung) {  
5             System.out.println("- " + pengunjung);  
6         }  
7     }  
8     public static void hargaJual (Integer... bayar) {  
9         System.out.println("Daftar Pembelian: ");  
10        for (int harga : bayar) {  
11            System.out.println("Rp " + harga);  
12        }  
13    }  
14 }  
Run main | Debug main  
15 public static void main(String[] args) {  
16     daftarPengunjung("Ali", "Budi", "Citra");  
17     hargaJual(15000, 20000, 22000);  
18 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL Filter (e.g. text)

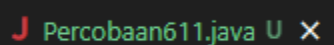
```
Daftar nama Pengunjung:  
- Ali  
- Budi  
- Citra  
Daftar Pembelian:  
Rp 15000  
Rp 20000  
Rp 22000  
>  
pleted. Ln 16, Col 50 Spaces: 4
```



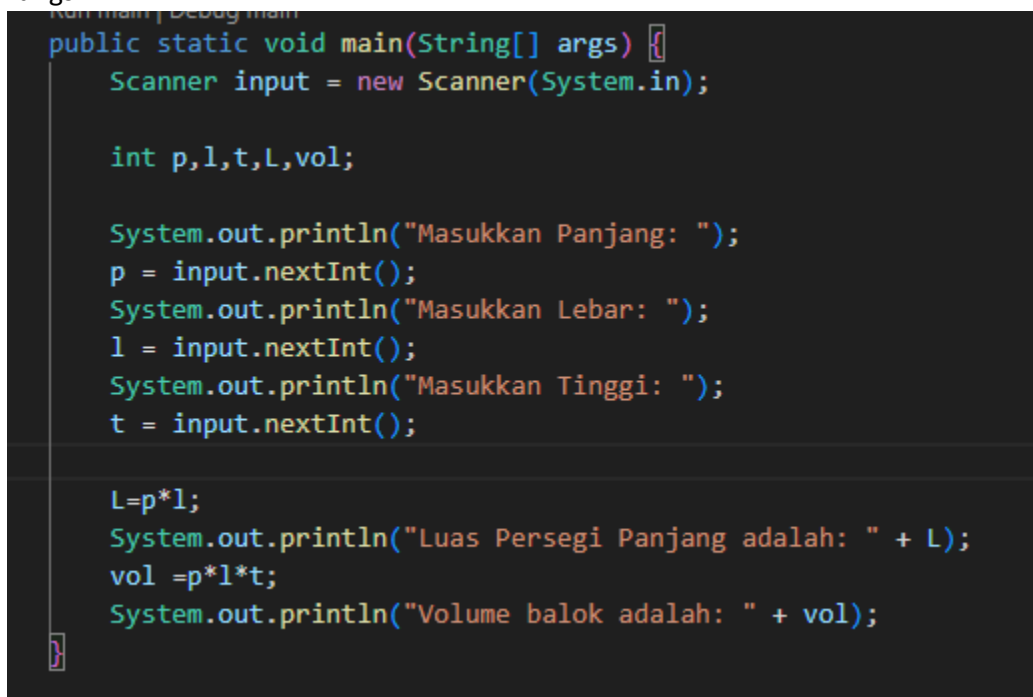
```
1 public class PengunjungCafe11 {
2     public static void daftarPengunjung (String... namaPengunjung, String... bayar) {
3         System.out.println("Daftar nama Pengunjung: ");
4         for (String pengunjung : namaPengunjung) {
5             System.out.println("- " + pengunjung);
6         }
7     }
8
9     public static void main(String[] args) {
10         daftarPengunjung("Ali", "Budi", "Citra");
11     }
12 }
```

Percobaan 5

1. Buat class baru, simpan file tersebut dengan nama Percobaan6NoAbsen.java



2. Buatlah program untuk menghitung luas persegi panjang dan volume balok tanpa menggunakan fungsi



```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    int p,l,t,L,vol;

    System.out.println("Masukkan Panjang: ");
    p = input.nextInt();
    System.out.println("Masukkan Lebar: ");
    l = input.nextInt();
    System.out.println("Masukkan Tinggi: ");
    t = input.nextInt();

    L=p*l;
    System.out.println("Luas Persegi Panjang adalah: " + L);
    vol =p*l*t;
    System.out.println("Volume balok adalah: " + vol);
}
```

3. . Program menghitung luas persegi dan volume balok di atas jika dibuatkan fungsi maka terdapat 3 fungsi yaitu hitungLuas, hitungVolume dan fungsi main, seperti di bawah ini

```

import java.util.Scanner;
public class Percobaan611 {
    public static int hitungLuas (int pjg, int lb) {
        int Luas = pjg*lb;
        return Luas;
    }
    public static int hitungVolume (int tinggi, int a, int b) {
        int volume = hitungLuas(a,b) * tinggi;
        return volume;
    }
    Run main | Debug main
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

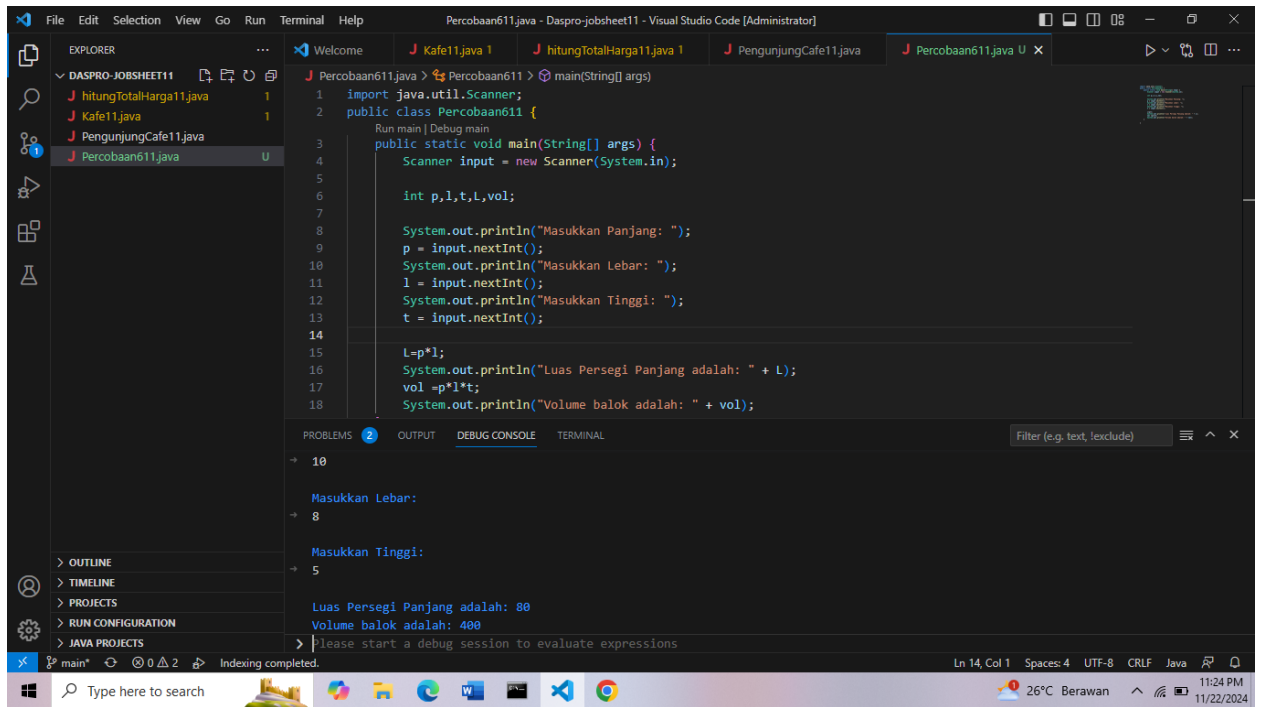
        int p,l,t,L,vol;

        System.out.println("Masukkan Panjang: ");
        p = input.nextInt();
        System.out.println("Masukkan Lebar: ");
        l = input.nextInt();
        System.out.println("Masukkan Tinggi: ");
        t = input.nextInt();

        L=hitungLuas(p,l);
        System.out.println("Luas Persegi Panjang adalah: " + L);
        vol =hitungVolume(t, p, l);
        System.out.println("Volume balok adalah: " + vol);
    }
}

```

4. Lakukan validasi hasil dengan memberikan contoh input pada program tersebut! Tampilkan hasilnya!

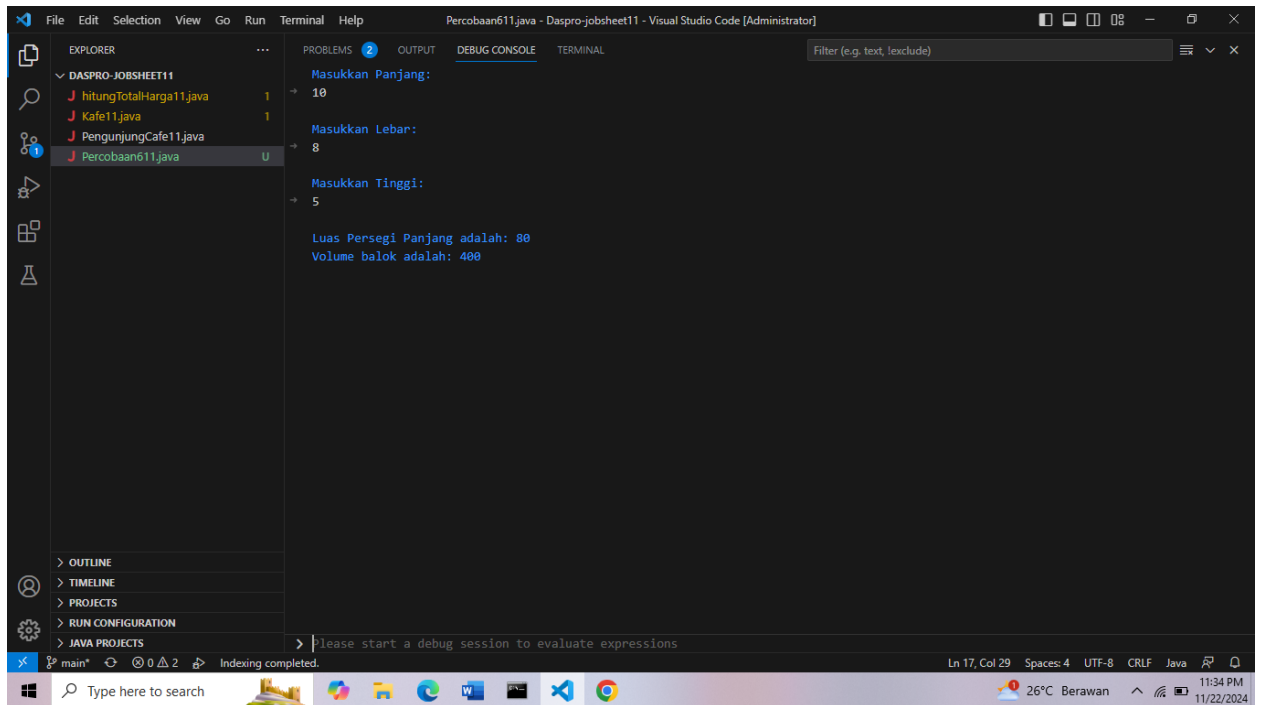


```
import java.util.Scanner;
public class Percobaan611 {
    public static int hitungLuas (int pjg, int lb) {
        int Luas = pjg*lb;
        return Luas;
    }
    public static int hitungVolume (int tinggi, int a, int b) {
        int volume = hitungLuas(a,b) * tinggi;
        return volume;
    }
    Run main | Debug main
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int p,l,t,L,vol;

        System.out.println("Masukkan Panjang: ");
        p = input.nextInt();
        System.out.println("Masukkan Lebar: ");
        l = input.nextInt();
        System.out.println("Masukkan Tinggi: ");
        t = input.nextInt();

        L=hitungLuas(p,l);
        System.out.println("Luas Persegi Panjang adalah: " + L);
        vol =hitungVolume(t, p, l);
        System.out.println("Volume balok adalah: " + vol);
    }
}
```



Jawaban Pertanyaan:

1. Sebutkan tahapan dan urutan eksekusi pada percobaan 6 di atas!

Tahapan dan urutan eksekusi

1. Deklarasi dan Definisi Fungsi:
 - a. Program dimulai dengan mendeklarasikan dua buah fungsi, yaitu:
 - a. `hitungLuas(int pjg, int lb)` untuk menghitung luas persegi panjang.
 - b. `hitungVolume(int tinggi, int a, int b)` untuk menghitung volume balok berdasarkan luas dasar dan tinggi.
2. Eksekusi fungsi main:

Ketika program dijalankan, eksekusi dimulai pada metode `main(String[] args)`.
3. Input dari Pengguna:
 - a. Program meminta input dari pengguna untuk panjang (`p`), lebar (`l`), dan tinggi (`t`):
 - a. `System.out.println("Masukkan Panjang: ");` untuk meminta panjang.
 - b. `System.out.println("Masukkan Lebar: ");` untuk meminta lebar.
 - c. `System.out.println("Masukkan Tinggi: ");` untuk meminta tinggi.
 - d. Pengguna diminta untuk memasukkan nilai panjang, lebar, dan tinggi yang kemudian disimpan dalam variabel `p`, `l`, dan `t`.
4. Panggilan Fungsi `hitungLuas`:
 - a. Setelah mendapatkan input panjang (`p`) dan lebar (`l`), program memanggil fungsi `hitungLuas(p, l)` untuk menghitung luas persegi panjang:

- a. Fungsi hitungLuas menerima dua parameter (pgj dan lb), yaitu panjang dan lebar yang diberikan oleh pengguna.
- b. Fungsi ini mengalikan panjang (pgj) dengan lebar (lb) dan menghasilkan nilai luas yang disimpan dalam variabel L.
- c. Setelah itu, hasil luas dicetak ke layar dengan `System.out.println("Luas Persegi Panjang adalah: " + L);`.

5. panggilan Fungsi hitungVolume:

- a. Setelah menghitung luas, program melanjutkan dengan memanggil fungsi `hitungVolume(t, p, l)` untuk menghitung volume balok:
 - a. Fungsi `hitungVolume` menerima tiga parameter, yaitu tinggi (tinggi), panjang (a), dan lebar (b).
 - b. Fungsi `hitungVolume` memanggil `hitungLuas(a, b)` untuk menghitung luas dasar balok terlebih dahulu (menggunakan panjang dan lebar).
 - c. Kemudian, luas dasar dikalikan dengan tinggi (tinggi) untuk mendapatkan volume balok.
 - d. Hasil volume disimpan dalam variabel `vol`.
 - e. Setelah itu, volume dicetak ke layar dengan `System.out.println("Volume balok adalah: " + vol);`.

6. Program Selesai:

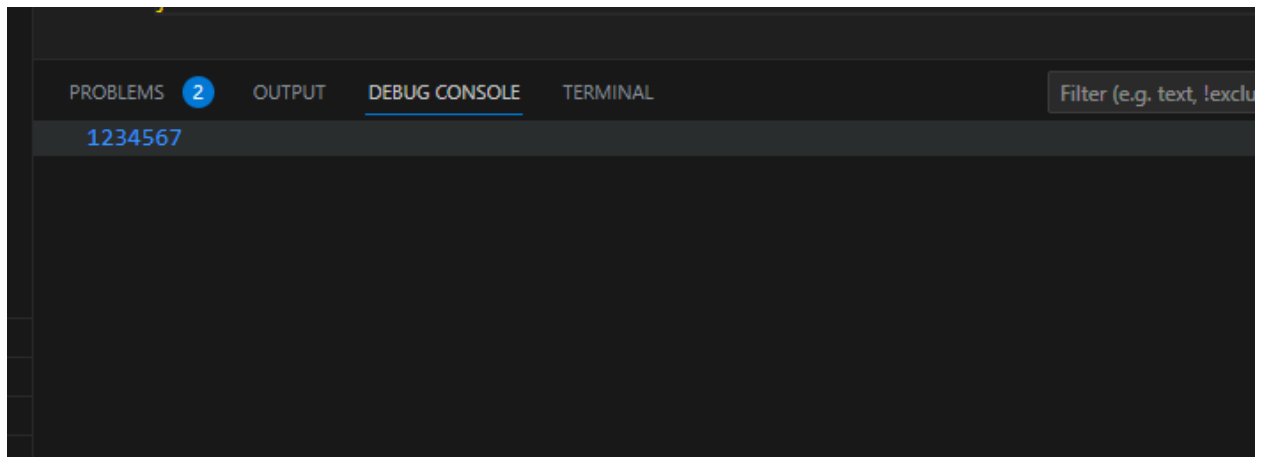
program selesai setelah mencetak hasil luas dan volume balok.

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

```

J programku.java > programku
1  public class programku {
2      public static void TampilHinggaKei (int i) {
3          for (int j = 1; j <= i; j++) {
4              System.out.print(j);
5          }
6      }
7      public static int Jumlah (int bil1, int bil2) {
8          return (bil1 + bil2);
9      }
10     public static void TampilJumlah (int bil1, int bil2) {
11         TampilHinggaKei(Jumlah (bil1,bil2));
12     }
13     Run main | Debug main
14     public static void main(String[] args) {
15         int temp = Jumlah(1, 1);
16         TampilJumlah (temp, 5);
17     }

```



Alur jalan nya program:

- **Langkah 1: Eksekusi fungsi main**

- Program mulai dieksekusi pada metode `main(String[] args)`.
- Pada baris pertama dalam main, program memanggil metode `Jumlah(1, 1)` dan menyimpan hasilnya dalam variabel `temp`.

- **Langkah 2: Eksekusi Jumlah (1, 1)**

- Metode `Jumlah` menerima dua argumen `bil1 = 1` dan `bil2 = 1`.
- Dalam metode `Jumlah`, kedua bilangan tersebut dijumlahkan:
 - $bil1 + bil2 = 1 + 1 = 2$
- Program mengembalikan hasil 2 dan menyimpannya dalam variabel `temp`.

- **Langkah 3: Eksekusi TampilJumlah(temp, 5)**

- Nilai dari `temp` adalah 2, jadi program memanggil `TampilJumlah(2, 5)`.
- Di dalam `TampilJumlah`, program pertama-tama memanggil `Jumlah(2, 5)` untuk menghitung hasil penjumlahannya.

- **Langkah 4: Eksekusi Jumlah(2, 5)**

- Metode `Jumlah` menerima `bil1 = 2` dan `bil2 = 5`.
- Hasil penjumlahan: $2 + 5 = 7$.
- Program mengembalikan hasil 7.

- **Langkah 5: Eksekusi TampilHinggaKei(7)**

- Setelah memperoleh hasil penjumlahan 7 dari `Jumlah(2, 5)`, program memanggil metode `TampilHinggaKei(7)`.
- Di dalam `TampilHinggaKei`, program mencetak angka mulai dari 1 hingga 7 secara berurutan, yaitu: 1234567.

- **Langkah 6: Program Selesai**

- Setelah menampilkan angka dari 1 hingga 7, program selesai dan keluar
3. 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!

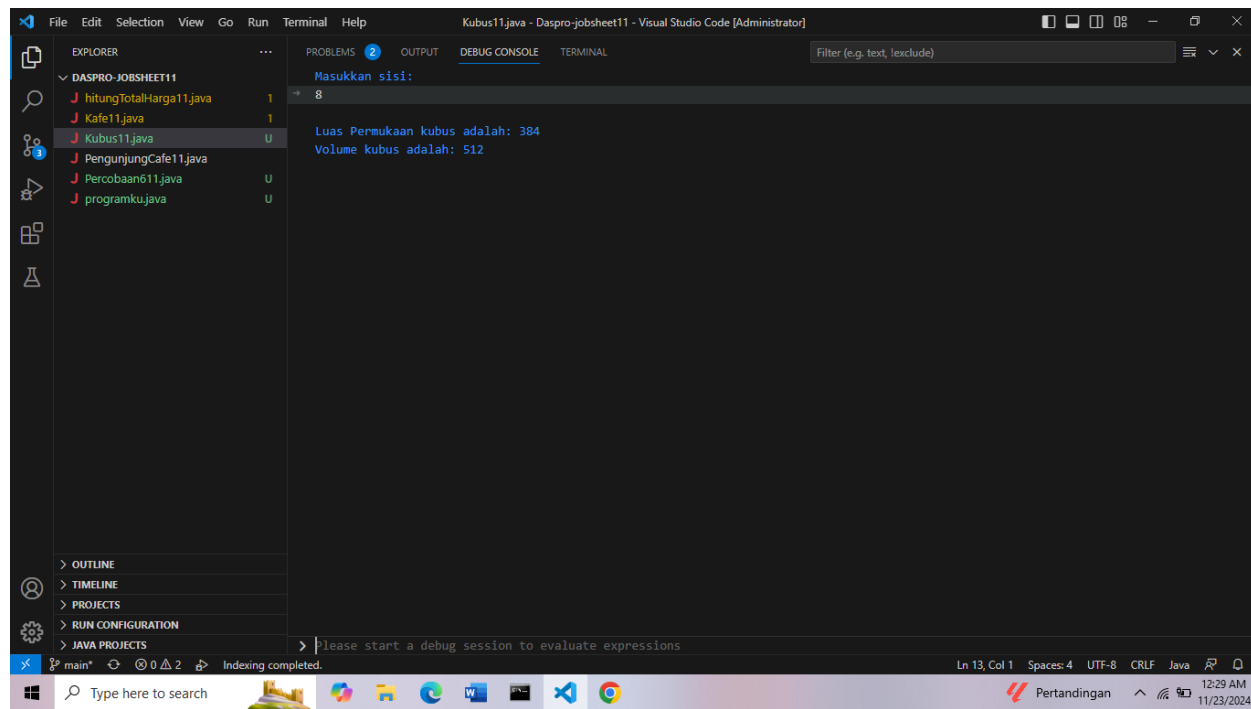
Parameter dibutuhkan pada fungsi ketika fungsi perlu menerima **input atau data tambahan** untuk diproses.

Nilai kembalian (return) dibutuhkan ketika fungsi **menghasilkan suatu nilai** yang perlu digunakan atau diproses lebih lanjut dalam program. Fungsi yang tidak membutuhkan nilai kembalian, seperti yang hanya bertugas untuk menampilkan pesan atau memperbarui data, tidak memerlukan return dan menggunakan void.

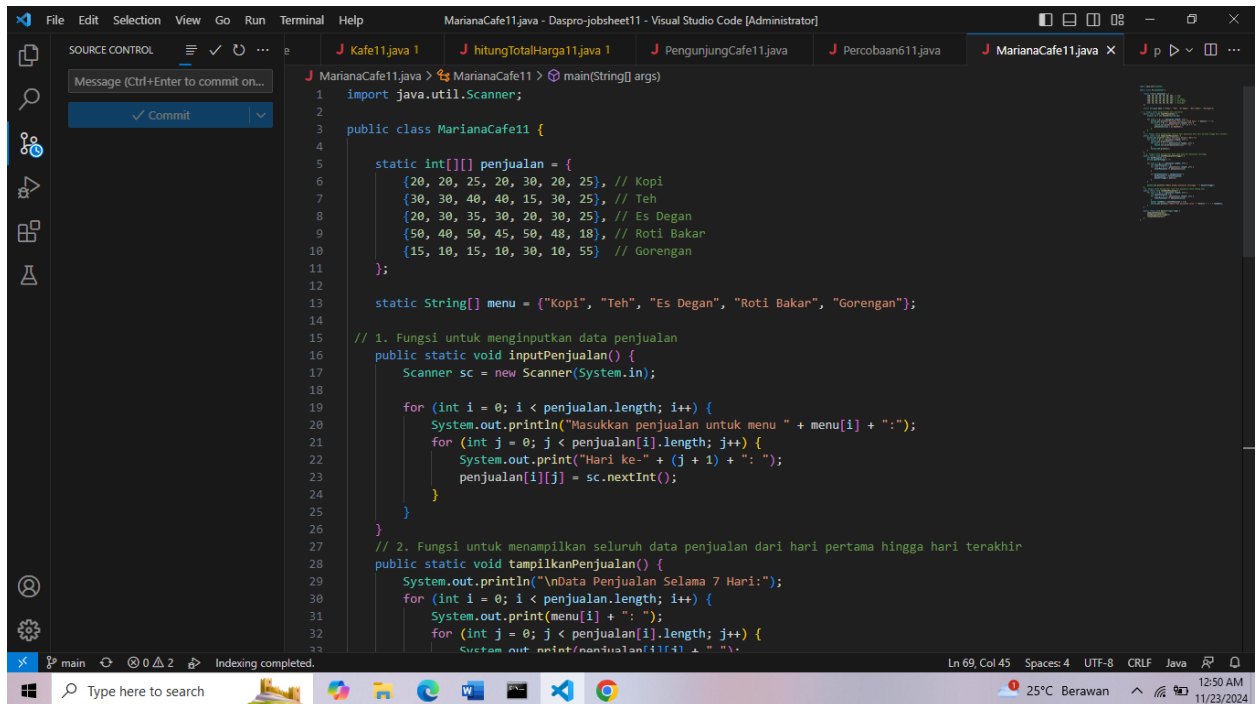
TUGAS

1. Buatlah sebuah class KubusNoAbsen yang di dalamnya terdapat fungsi untuk menghitung volume kubus dan luas permukaan kubus

```
J Kubus11.java > Kubus11 > main(String[] args)
1  import java.util.Scanner;
2  public class Kubus11 {
3      public static int hitungVolume (int sisi) {
4          int volume = sisi * sisi * sisi;
5          return volume;
6      }
7      public static int LuasPermukaan (int sisi) {
8          int LuasPermukaan = 6 * sisi * sisi;
9          return LuasPermukaan;
10     }
11     public static void main(String[] args) {
12         Scanner input = new Scanner(System.in);
13
14         System.out.println("Masukkan sisi: ");
15         int sisi = input.nextInt();
16
17         int LP = hitungLuasPermukaan(sisi);
18         System.out.println("Luas Permukaan kubus adalah: " + LP );
19         int vol = hitungVolume(sisi);
20         System.out.println("Volume kubus adalah: " + vol);
21     }
22 }
```



2. . Ibu Mariana mengajar café. Berikut adalah rekap penjualan 5 menu dari hari pertama hingga ketujuh:



The screenshot shows the Visual Studio Code editor with the file MarianaCafe11.java open. The code implements two methods: menuPenjualanTertinggi() and rataRataPenjualan(). The menuPenjualanTertinggi() method iterates through the penjualan array, calculates the totalPenjualan for each menu item, and updates maxPenjualan and menuTertinggi if a higher total is found. The rataRataPenjualan() method calculates the average price (rataRata) for each menu item by dividing the totalPenjualan by 7.0. The status bar at the bottom indicates the current position is Ln 69, Col 45, with 4 spaces, UTF-8 encoding, and CRLF line endings.

```
33     System.out.print(penjualan[i][j] + " ");
34     }
35     System.out.println();
36 }
37 }
38 // 3. Fungsi untuk menampilkan Menu yang memiliki penjualan tertinggi
39 public static void menuPenjualanTertinggi() {
40     int maxPenjualan = 0;
41     String menuTertinggi = "";
42
43     for (int i = 0; i < penjualan.length; i++) {
44         int totalPenjualan = 0;
45         for (int j = 0; j < penjualan[i].length; j++) {
46             totalPenjualan += penjualan[i][j];
47         }
48
49         if (totalPenjualan > maxPenjualan) {
50             maxPenjualan = totalPenjualan;
51             menuTertinggi = menu[i];
52         }
53     }
54
55     System.out.println("\nMenu dengan penjualan tertinggi: " + menuTertinggi);
56 }
57 //4. Fungsi untuk menampilkan rata-rata penjualan untuk setiap menu
58 public static void rataRataPenjualan() {
59     for (int i = 0; i < penjualan.length; i++) {
60         int totalPenjualan = 0;
61         for (int j = 0; j < penjualan[i].length; j++) {
62             totalPenjualan += penjualan[i][j];
63         }
64         double rataRata = totalPenjualan / 7.0;
```

The screenshot shows the continuation of the MarianaCafe11.java file in Visual Studio Code. It includes the implementation of the rataRataPenjualan() method, which prints the average price for each menu item. Below this, the main() method is implemented, which calls the tampilkanPenjualan(), menuPenjualanTertinggi(), and rataRataPenjualan() methods. The status bar at the bottom indicates the current position is Ln 69, Col 45, with 4 spaces, UTF-8 encoding, and CRLF line endings.

```
65         System.out.println("\nRata-rata penjualan untuk " + menu[i] + ": " + rataRata);
66     }
67 }
68
69 Run main | Debug main
70 public static void main(String[] args) {
71     tampilkanPenjualan();
72     menuPenjualanTertinggi();
73     rataRataPenjualan();
74 }
75 }
```

File Edit Selection View Go Run Terminal Help

MarianaCafe11.java - Daspro-jobsheet11 - Visual Studio Code [Administrator]

EXPLORER

▼ DASPRO-JOBSHEET11

J hitungTotalHarga11.java

J Kafe11.java

J Kubus11.java

J MarianaCafe11.java

J PengunjungCafe11.java

J Percobaan611.java

J programkujava

PROBLEMS

2

OUTPUT

DEBUG CONSOLE

TERMINAL

Filter (e.g. text, exclude)

Ln 69, Col 45

Spaces: 4

UTF-8

CRLF

Java

main*

0 2

Indexing completed.

25°C

Berawan

12:49 AM

11/23/2024

```
Data Penjualan Selama 7 Hari:  
Kopi: 20 20 25 20 30 20 25  
Teh: 30 30 40 40 15 30 25  
Es Degan: 20 30 35 30 20 30 25  
Roti Bakar: 50 40 50 45 50 48 18  
Gorengan: 15 10 15 10 30 10 55  
  
Menu dengan penjualan tertinggi: Roti Bakar  
  
Rata-rata penjualan untuk Kopi: 22.857142857142858  
  
Rata-rata penjualan untuk Teh: 30.0  
  
Rata-rata penjualan untuk Es Degan: 27.142857142857142  
  
Rata-rata penjualan untuk Roti Bakar: 43.0  
  
Rata-rata penjualan untuk Gorengan: 20.714285714285715
```