



JOB SHEET 12

Fungsi 1

1. Tujuan

1. Mahasiswa mampu memahami penggunaan fungsi *static* pada Java dengan parameter dan mengembalikan nilai.
2. Mahasiswa mampu membuat program menggunakan fungsi *static* dan mengeksekusi fungsi tersebut.

2. Praktikum

2.1 Percobaan 1: Membuat Fungsi Tanpa Parameter Waktu

Percobaan: 40 menit

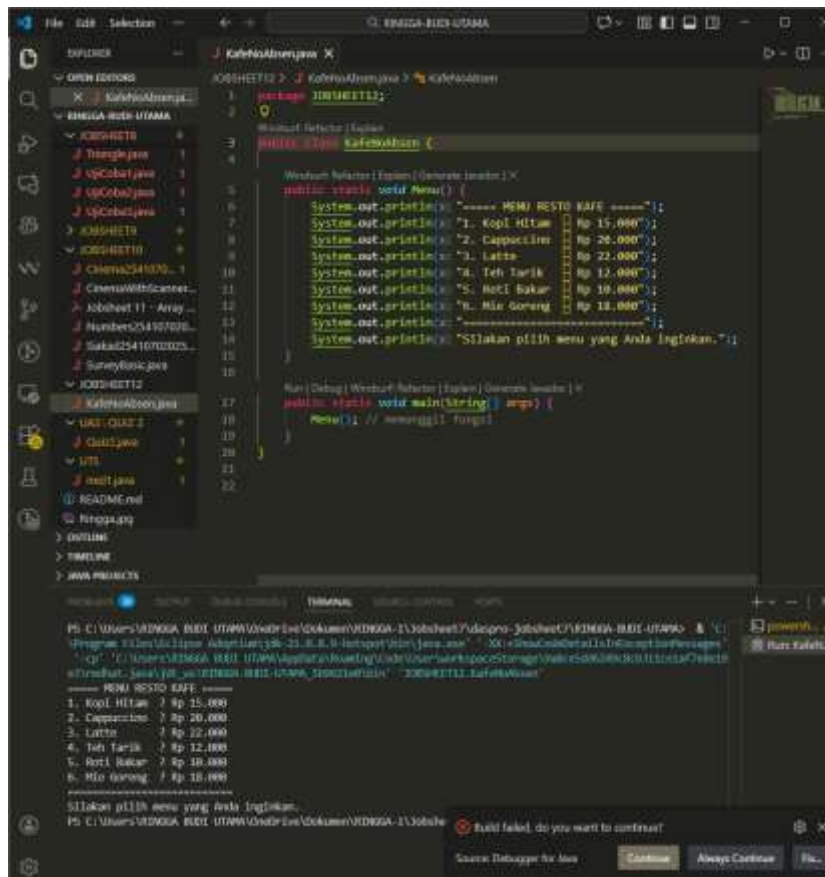
1. Buat *package* baru dengan nama **Pertemuan14** selanjutnya buat *class* baru, dan simpan file tersebut dengan nama **KafeNoAbsen.java**.
2. Buat fungsi **Menu** di dalam *class* tersebut.

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

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

```
1 public static void main(String[] args) {  
2     Menu();  
3 }
```

4. Push dan commit ke git.



```

package J08040212;

import java.util.Scanner;

public class KafeWahana {

    Window Refactor | Editor | Generate | Run | X
    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("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("=====");
    }

    Run | Debug | Window | Refactor | Editor | Generate | Run | X
    public static void main(String[] args) {
        Menu(); // memanggil fungsi
    }
}

```

Output:

```

===== 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.

```

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).



Jawaban :

- Fungsi tanpa parameter tidak harus bertipe `void`, karena fungsi tanpa parameter tetap boleh mengembalikan nilai (`return`).
- Daftar menu tetap dapat ditampilkan tanpa fungsi dengan menuliskan perintah `System.out.println()` langsung di dalam `main`.
- Keuntungan menggunakan fungsi `Menu()` adalah kode lebih rapi, mudah dibaca, mudah diubah, bisa dipanggil berulang tanpa menulis ulang perintah, dan memudahkan pemeliharaan program.
- Alur eksekusinya: program dimulai dari `main`, lalu ketika `Menu()` dipanggil program berpindah ke fungsi tersebut, perintah di dalamnya dijalankan hingga selesai, kemudian kembali lagi ke `main` dan melanjutkan program.

2.2 Percobaan 2: Membuat Fungsi Dengan Parameter

Waktu Percobaan: 40 menit

1. Buka kembali class **KafeNoAbsen.java** yang sudah dibuat pada Percobaan 1.
2. Ubah fungsi **Menu** dengan dua buah parameter bertipe *String* dan boolean.

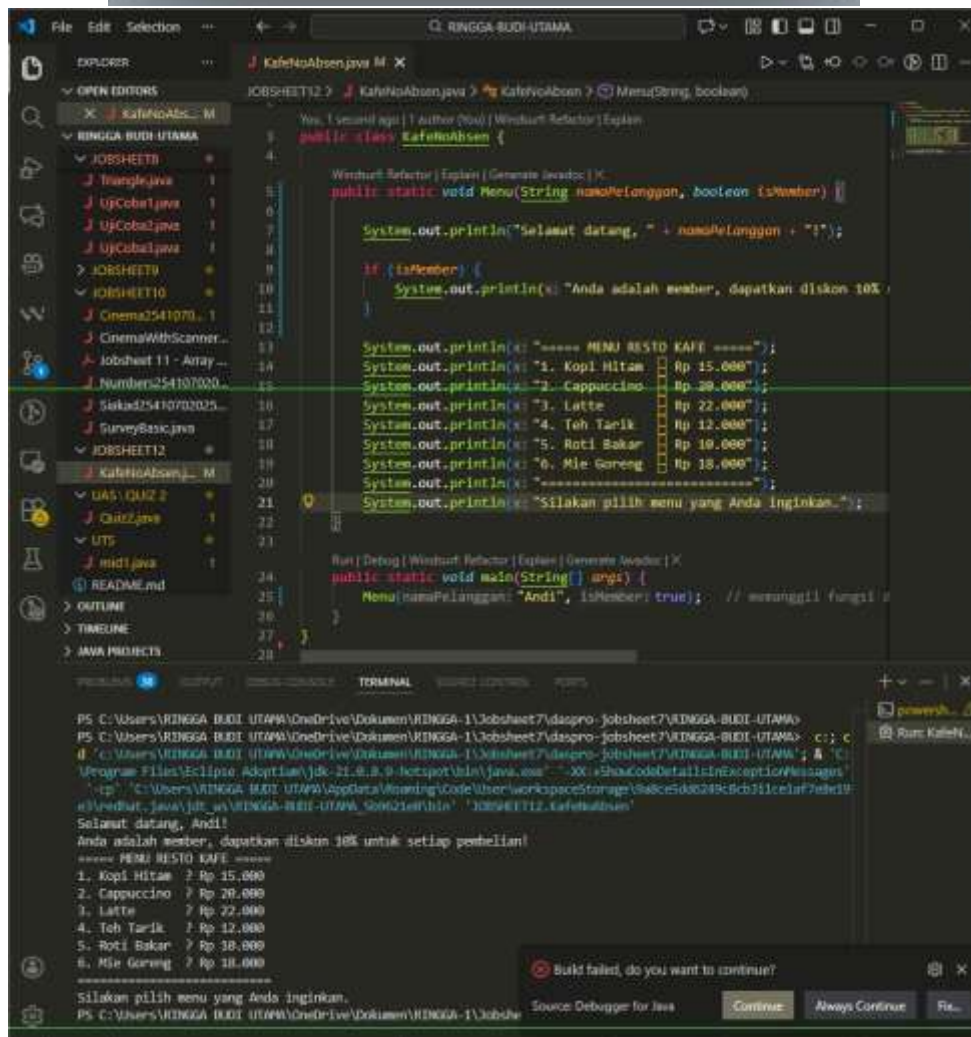
```
1 public static void Menu(String namaPelanggan, boolean isMember) {
2     System.out.println("Selamat datang, " + namaPelanggan + "!");
3
4     if (isMember) {
5         System.out.println("Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");
6     }
7
8     System.out.println("==== 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("3. Latte - Rp 22,000");
12    System.out.println("4. Teh Tarik - Rp 12,000");
13    System.out.println("5. Roti Bakar - Rp 10,000");
14    System.out.println("6. Mie Goreng - Rp 18,000");
15    System.out.println("=====");
16    System.out.println("Silakan pilih menu yang Anda inginkan.");
17 }
```

3. Eksekusi fungsi **Menu()** dari dalam fungsi *main*.

```

1 public static void main(String[] args) {
2     Menu("Andi", true);
3
4 }

```



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.

Jawaban :

1. Parameter digunakan untuk mengirim nilai dari luar fungsi agar fungsi dapat bekerja sesuai data yang diberikan.
2. Parameter **namaPelanggan** dan **isMember** digunakan untuk menampilkan menu yang berbeda sesuai nama pelanggan dan status keanggotaan.
3. Parameter bukan variabel biasa, tetapi saat fungsi dijalankan parameter akan berperan sebagai variabel lokal yang menyimpan nilai yang dikirim saat pemanggilan fungsi.
4. Parameter **isMember** digunakan untuk menentukan apakah pelanggan mendapatkan perlakuan khusus. Jika true, tampilannya menandakan member (misalnya dapat diskon), jika false, tampilannya seperti pelanggan biasa tanpa fasilitas member.
5. Jika fungsi **Menu()** dipanggil tanpa parameter, akan terjadi error karena program membutuhkan nilai untuk mengisi parameter tersebut.
6. Contoh modifikasi fungsi: **Menu(String namaPelanggan, boolean isMember, String kodePromo)** dan di dalamnya cek: jika **kodePromo** = "DISKON50" tampilkan diskon 50%, jika "DISKON30" tampilkan diskon 30%, jika selain itu tampilkan invalid.
7. Contoh pemanggilan fungsi yang benar: **Menu("Budi", true, "DISKON30");**
8. Ya, penggunaan parameter membuat program lebih mudah dibaca dan dikembangkan karena fungsi menjadi fleksibel, tidak perlu mengubah isi fungsi setiap kali data berubah, dan lebih mudah digunakan kembali.



2.3 Percobaan 3: Membuat Fungsi dengan Nilai Kembalian

Waktu Percobaan: 40 menit

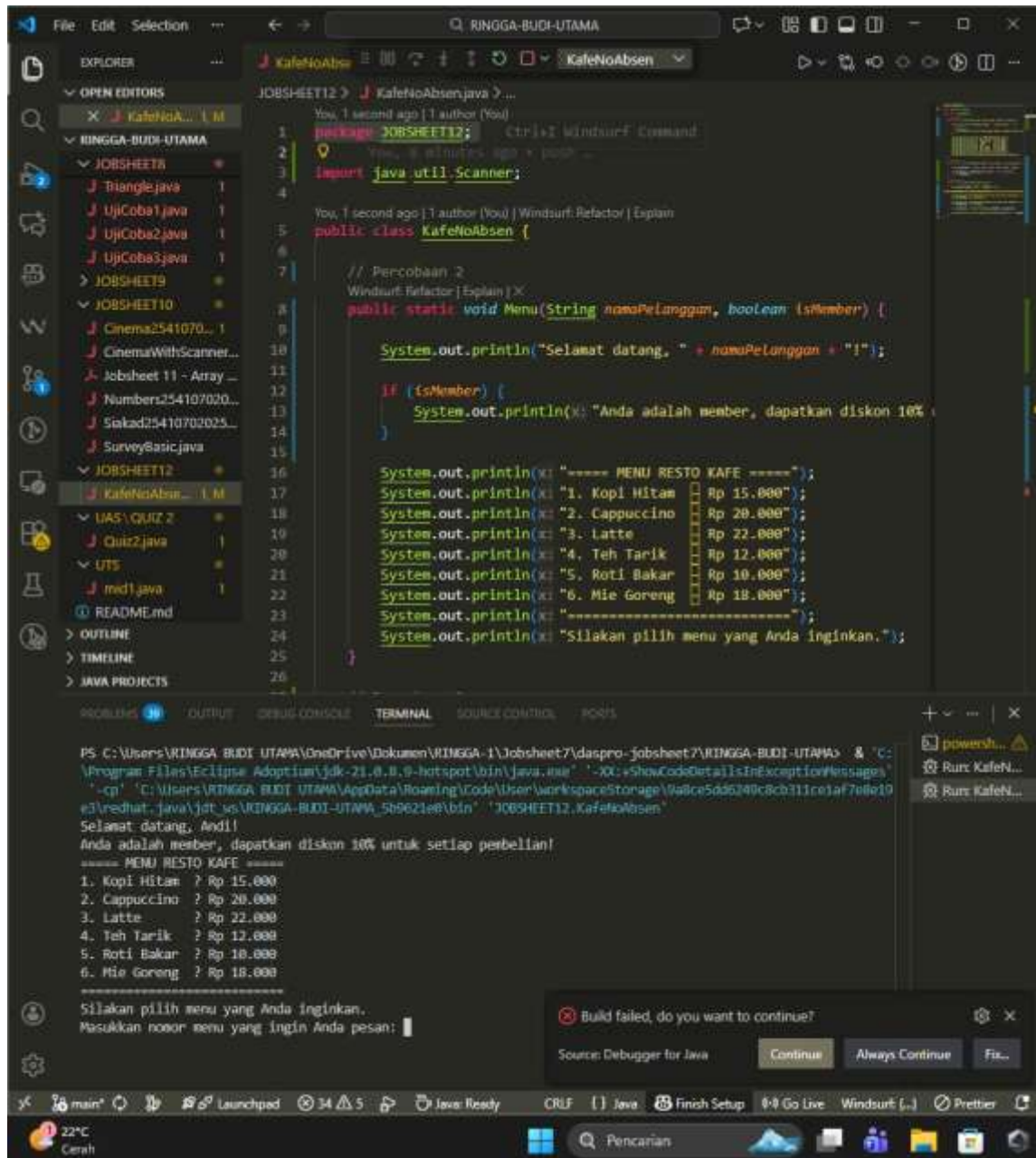
1. Buka kembali `class KafeNoAbsen.java` yang sudah digunakan pada Percobaan 1 dan 2.
2. Pada `class` yang sama buat fungsi `hitungTotalHargaNoAbsen` di dalam `class` tersebut yang mengembalikan nilai total harga dan parameter masukan pilihan Menu dan banyaknya pesanan. Tambahkan daftar harga untuk setiap menu kafe dalam variabel berupa Array.

```
1 public static int hitungTotalHarga(int pilihanMenu, int banyakItem) {  
2     int[] hargaItems = {15000, 20000, 22000, 12000, 10000, 18000};  
3  
4     int hargaTotal = hargaItems[pilihanMenu - 1] * banyakItem;  
5     return hargaTotal;  
6 }
```

3. Modifikasi fungsi `main` di dalam `class` tersebut, dan eksekusi fungsi `Menu()` dari dalam fungsi `main` untuk menampilkan daftar menu kafe.
4. Selanjutnya modifikasi fungsi `main` sehingga program dapat meminta input dari pengguna untuk `pilihanMenu`, `banyakItem` (jumlah porsi yang dipesan). Fungsi `main` juga memanggil fungsi `hitungTotalHargaNoAbsen` (`pilihanMenu`, `banyakItem`) dan menyimpan nilai kembalian (return value) ke dalam variabel `totalBayar`, lalu program menampilkan `totalBayar` ke layar.

```
1  
2 System.out.print("\nMasukkan nomor menu yang ingin Anda pesan: ");  
3 int pilihanMenu = sc.nextInt();  
4 System.out.print("Masukkan jumlah item yang ingin dipesan: ");  
5 int banyakItem = sc.nextInt();  
6  
7 int totalHarga = hitungTotalHarga(pilihanMenu, banyakItem);  
8  
9 System.out.println("Total harga untuk pesanan Anda: Rp" + totalHarga);
```

- Jalankan program dan amati apakah nilai total yang ditampilkan sudah sesuai dengan harga dan jumlah pesanan.



```

package JOBSHEET12;
import java.util.Scanner;

// Percobaan 2
public class KafeNoAbsen {

    public static void Menu(String namaPelanggan, boolean isMember) {

        System.out.println("Selamat datang, " + namaPelanggan + "!");

        if (isMember) {
            System.out.println("Anda adalah member, dapatkan diskon 10%");
        }

        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.");
    }
}
  
```

Build failed, do you want to continue?

Source: Debugger for Java

Continue Always Continue Fix...

Pertanyaan

- 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.
- Fungsi **hitungTotalHargaNoAbsen** saat ini mengembalikan total harga berdasarkan **pilihanMenu** dan **jumlahPesanan**. Sebutkan tipe data nilai kembalian dan dua



buah parameter yang digunakan fungsi tersebut. Jelaskan arti masing-masing parameter dalam konteks program kafe.

3. Modifikasi kode di atas sehingga fungsi **hitungTotalHargaNoAbsen** dapat menerima **kodePromo**. Jika **kodePromo** adalah "DISKON50", maka mendapat diskon 50% dari **totalHarga** dan tampilkan diskon. Jika **kodePromo** adalah "DISKON30", maka mendapat diskon 30% dari **totalHarga** dan tampilkan diskon. Jika tidak ada kode promo yang berlaku, tampilkan kode invalid dan tidak ada pengurangan total harga **totalHarga**.
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 :

- **Fungsi membutuhkan return value** ketika kita perlu membawa hasil perhitungan kembali ke pemanggil. Contoh: `hitungTotalHargaNoAbsen()` mengembalikan total harga.

Tidak perlu return value jika hanya menampilkan output. Contoh: `Menu()` hanya menampilkan menu.

- `hitungTotalHargaNoAbsen` memiliki **tipe data nilai kembalian int**. Parameternya ada 2:

`int pilihanMenu` → menentukan harga dari menu yang dipilih.

`int jumlahPesanan` → menentukan berapa banyak pesanan sehingga berpengaruh ke total harga.

- Modifikasi singkat logika untuk menerima promo:

Jika DISKON50 → diskon 50% dari total.

Jika DISKON30 → diskon 30% dari total.

Jika selain itu → tampil "invalid" dan total tetap.

- Modifikasi agar dapat memesan beberapa menu dan menghitung total semua pesanan:

Tambahkan **perulangan** untuk input pesanan lebih dari satu menu.

Setiap menu yang dipilih dihitung memakai fungsi `hitungTotalHargaNoAbsen`.

Gunakan **variabel accumulator** seperti `totalKeseluruhan += totalPerMenu`.

Setelah semua menu dipilih, tampilkan `totalKeseluruhan`.



2.4 Percobaan 4: Fungsi Varargs Waktu

Percobaan: 40 menit

1. Buat *class* baru, simpan file tersebut dengan nama **PengunjungCafeNoAbsen.java**.
2. Buatlah fungsi **daftarPengunjung** (bertipe void) di dalam *class* tersebut dengan menggunakan parameter varArgs

```
static void daftarPengunjung(String...namaPengunjung) {  
    System.out.println("Daftar Nama Pengunjung:");  
    for(int i = 0; i < namaPengunjung.length; i++){  
        System.out.println("- "+namaPengunjung[i]);  
    }  
}
```

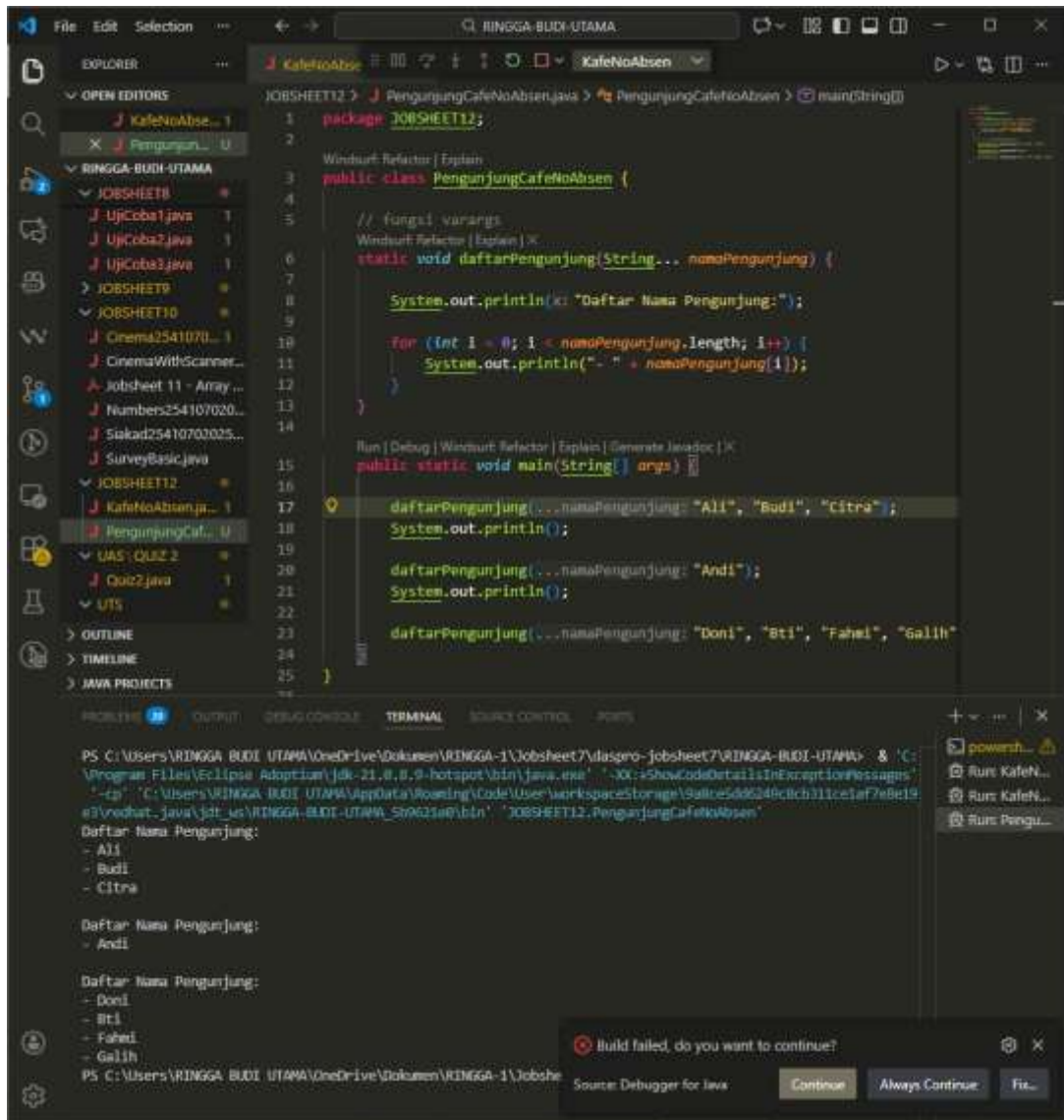
3. Buat fungsi **main** di dalam *class* tersebut, dan eksekusi fungsi namaPengunjung dari dalam fungsi *main*.

```
daftarPengunjung("Ali", "Budi", "Citra");
```

4. Jalankan program dan amati hasilnya
5. Tambahkan dua baris berikut pada fungsi main di bawah kode pada nomor 3 di atas:

```
daftarPengunjung("Andi");  
daftarPengunjung("Doni", "Eti", "Fahmi", "Galih");
```

6. Jalankan program. Amati perbedaan output ketika jumlah argumen yang dikirim berbeda-beda.



```

package JOBSHEET12;

public class PengunjungCafeNoAbsen {

    // fungsi varargs
    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) {

        daftarPengunjung(...namaPengunjung= "Ali", "Budi", "Citra");
        System.out.println();

        daftarPengunjung(...namaPengunjung= "Andi");
        System.out.println();

        daftarPengunjung(...namaPengunjung= "Doni", "Bti", "Fahel", "Galih")
    }
}
    
```

PS C:\Users\RINGGA_BUDI_UTAMA\OneDrive\Documents\RINGGA-1\Jobsheet7\ulaspro-Jobsheet7\RINGGA-BUDI-UTAMA> & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.8-hotspot\bin\java.exe' -XX:+ShowCodeDetailsInExceptionMessages -cp 'C:\Users\RINGGA_BUDI_UTAMA\AppData\Local\Temp\Code\User\workspaceStorage\9a1ce5dd6246c8c311c1eaf7e8e19e3\vedhat_java\jdt_ws\RINGGA-BUDI-UTAMA_ShrAG21w@b1n' JOBSHEET12.PengunjungCafeNoAbsen

Daftar Nama Pengunjung:

- Ali

- Budi

- Citra

Daftar Nama Pengunjung:

- Andi

Daftar Nama Pengunjung:

- Doni

- Bti

- Fahel

- Galih

Build failed, do you want to continue?

Source: Debugger for Java

Continue Always Continue Fix...

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?



Jawaban :

1. Karena menggunakan varargs agar jumlah parameter fleksibel (String...).
2. Gunakan for-each:
for(String n : namaPengunjung){}
3. Tidak bisa dua varargs dalam satu fungsi. Varargs hanya boleh satu dan harus di posisi terakhir.
4. Tetap berjalan, tidak error. Varargs menjadi array kosong dan output hanya default method tanpa data.

2.5 Percobaan 5: Pembuatan Kode Program, dengan Fungsi versus Tanpa

Fungsi Waktu Percobaan: 50 menit

1. Buat *class* baru, simpan file tersebut dengan nama **HitungBalokNoAbsen.java**.
2. Buatlah program untuk menghitung luas persegi panjang dan volume balok tanpa menggunakan fungsi

```
import java.util.Scanner;

public class HitungBalok {
    Run | Debug
    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);

        input.close();
    }
}
```



- 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:

Fungsi hitungLuas

```
static int hitungLuas(int pjg, int lb) {
    int Luas = pjg * lb;
    return Luas;
}
```

Fungsi hitungVolume

```
static int hitungVolume(int tinggi, int a, int b) {
    int volume = hitungLuas(a, b) * tinggi;
    return volume;
}
```

Fungsi 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);
    input.close();
}
```

- Lakukan validasi hasil dengan memberikan contoh inputan pada program tersebut! Tampilkan hasilnya!



```

package JOBSHEET12;

public class HitungKubusNoAbsen {

    public static void main(String[] args) {

        java.util.Scanner input = new java.util.Scanner(System.in);

        int p, l, t, volume;

        System.out.print(s: "Masukkan panjang: ");
        p = input.nextInt();

        System.out.print(s: "Masukkan lebar: ");
        l = input.nextInt();

        System.out.print(s: "Masukkan tinggi: ");
        t = input.nextInt();

        volume = p * l * t;

        System.out.println("Volume kubus adalah = " + volume);
    }
}

```

Masukkan lebar: \20\

Exception in thread "main" java.util.InputMismatchException
 at java.base/java.util.Scanner.throwFor(Scanner.java:947)
 at java.base/java.util.Scanner.next(Scanner.java:1602)
 at java.base/java.util.Scanner.nextInt(Scanner.java:2267)
 at java.base/java.util.Scanner.nextInt(Scanner.java:2221)
 at JOBSHEET12.HitungKubusNoAbsen.main(HitungKubusNoAbsen.java:15)

PS C:\Users\RINGGA BUDI UTAMA\OneDrive\Dokumen\RINGGA-1\Jobsheet7\daspro-jobsheet7\RINGGA-BUDI-UTAMA>
 PS C:\Users\RINGGA BUDI UTAMA\OneDrive\Dokumen\RINGGA-1\Jobsheet7\daspro-jobsheet7\RINGGA-BUDI-UTAMA> ^C
 PS C:\Users\RINGGA BUDI UTAMA\OneDrive\Dokumen\RINGGA-1\Jobsheet7\daspro-jobsheet7\RINGGA-BUDI-UTAMA>
 PS C:\Users\RINGGA BUDI UTAMA\OneDrive\Dokumen\RINGGA-1\Jobsheet7\daspro-jobsheet7\RINGGA-BUDI-UTAMA> c:: c
 d 'c:\Users\RINGGA BUDI UTAMA\OneDrive\Dokumen\RINGGA-1\Jobsheet7\daspro-jobsheet7\RINGGA-BUDI-UTAMA'; & 'C:
 \Program Files\Eclipse Adoptium\jdk-21.0.8-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
 '-cp' 'C:\Users\RINGGA BUDI UTAMA\AppData\Roaming\Code\User\workspaceStorage\9a8ce5dd6249c8cb311ce1af7e8e19
 e3\redhat.java\jdt_ws\RINGGA-BUDI-UTAMA_5b9621e0\bin' 'JOBSHEET12.HitungKubusNoAbsen'

Masukkan panjang: 20
 Masukkan lebar: 6
 Masukkan tinggi: 5
 Volume kubus adalah = 600
 PS C:\Users\RINGGA BUDI UTAMA\OneDrive\Dokumen\RINGGA-1\Jobsheet7\daspro-jobsheet7\RINGGA-BUDI-UTAMA>

Pertanyaan

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



2. Tuliskan output yang dihasilkan program HitungBalokNoAbsen jika pengguna memasukkan panjang = 4, lebar = 3, dan tinggi = 5. Jelaskan secara singkat alur jalannya program sampai output tersebut muncul.
3. Apakah output dari program di bawah ini kemudian jelaskan alur jalannya program tersebut!

```

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
8     public static int Jumlah (int bil1, int bil2) {
9         return (bil1 + bil2);
10    }
11
12    public static void TampilJumlah (int bil1, int bil2) {
13        TampilHinggaKei(Jumlah(bil1, bil2));
14    }
15
16    public static void main (String[] args) {
17        int temp = Jumlah(1, 1);
18        TampilJumlah(temp, 5);
19    }
20 }

```

4. Pada saat apakah fungsi yang kita buat harus menggunakan parameter atau tidak? Pada saat apakah fungsi yang kita buat harus memiliki nilai kembalian atau tidak? Jelaskan!
5. Jelaskan kapan sebuah fungsi sebaiknya menggunakan parameter dan kapan fungsi boleh tanpa parameter, dengan mengacu pada fungsi **hitungLuas** dan **hitungVolume** di Percobaan 5.
6. Jelaskan kapan sebuah fungsi sebaiknya memiliki nilai kembalian (return value) dan kapan tidak perlu memiliki nilai kembalian, dengan mengacu pada fungsi **main**, **hitungLuas**, dan **hitungVolume**.

Jawaban :

1. Program mulai → main dipanggil → input panjang/lebar/tinggi → fungsi hitungLuas dipanggil → fungsi hitungVolume dipanggil → hasil dikembalikan → ditampilkan ke layar.
2. Input 4,3,5 → luas = $4 \times 3 = 12$ → volume = $4 \times 3 \times 5 = 60$ → output:
 - o Luas: 12
 - o Volume: 60
3. Output program adalah nilai luas dan volume sesuai perhitungan → alur: main → baca input → panggil fungsi → fungsi menghitung → hasil dikirim balik → dicetak.
4. Parameter diperlukan jika fungsi butuh data dari luar.
Tidak perlu parameter jika fungsi sudah punya data sendiri.

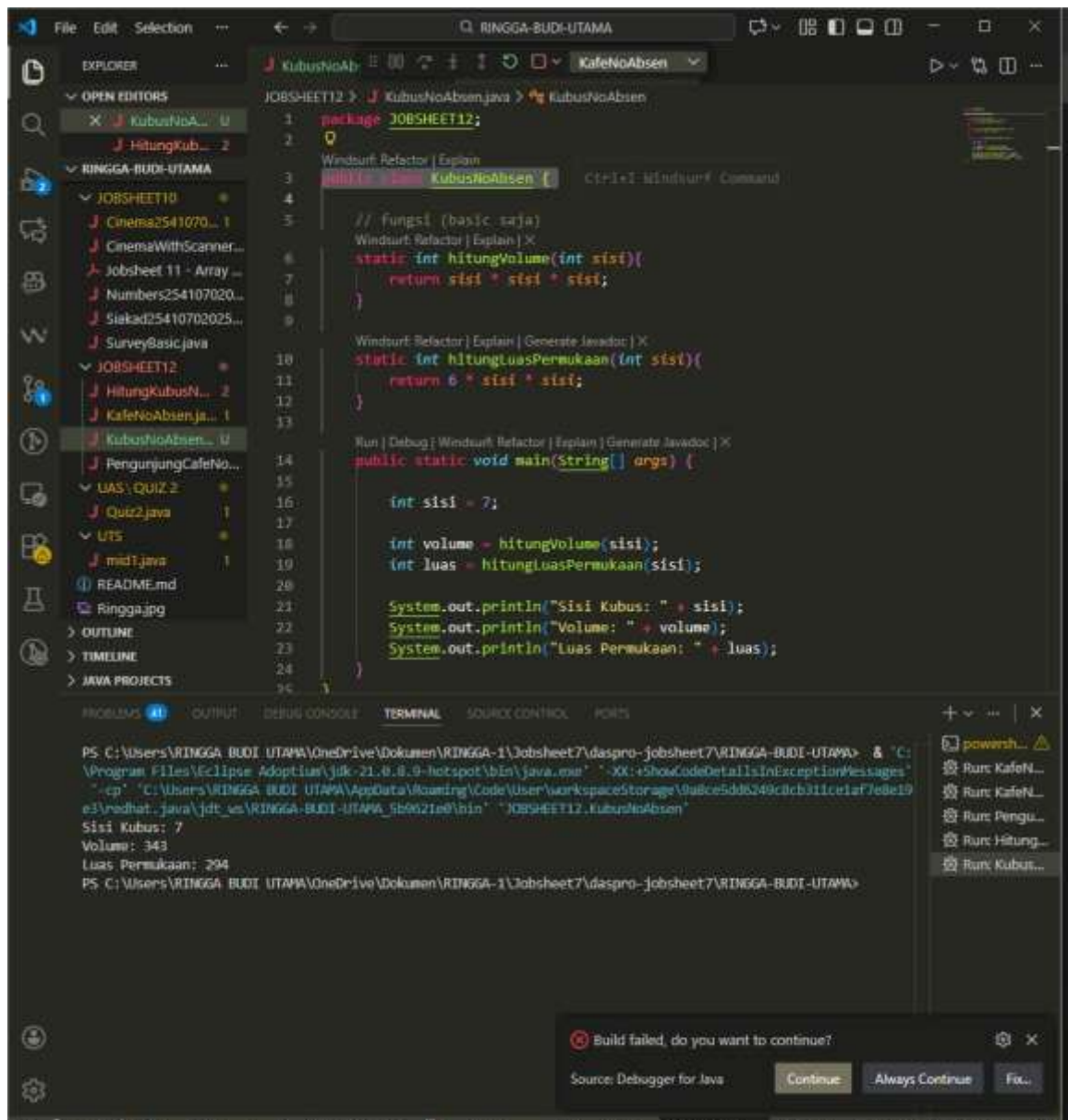


Return digunakan jika fungsi harus mengirim nilai balik. Tidak perlu return jika hanya menampilkan proses.

5. hitungLuas dan hitungVolume pakai parameter karena butuh panjang/lebar/tinggi dari main. Tidak perlu parameter kalau nilai sudah diketahui di fungsi.
6. hitungLuas dan hitungVolume butuh return karena menghasilkan nilai. main tidak butuh return karena hanya menjalankan program, bukan memberi nilai ke fungsi lain.

3. Tugas Waktu Pengerjaan: 100 menit

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



```

package JOBSHEET12;

public class KubusNoAbsen {

    // fungsi (basic saja)
    static int hitungVolume(int sisi){
        return sisi * sisi * sisi;
    }

    static int hitungLuasPermukaan(int sisi){
        return 6 * sisi * sisi;
    }

    public static void main(String[] args) {

        int sisi = 7;

        int volume = hitungVolume(sisi);
        int luas = hitungLuasPermukaan(sisi);

        System.out.println("Sisi Kubus: " + sisi);
        System.out.println("Volume: " + volume);
        System.out.println("Luas Permukaan: " + luas);
    }
}
    
```

PS C:\Users\RINGGA BUDI UTAMA\OneDrive\Dokumen\RINGGA-1\Jobsheet7\daspro-jobsheet7\RINGGA-BUDI-UTAMA> & "C:\Program Files\Eclipse Adoptium\jdk-21.0.8-hotspot\bin\java.exe" "-XX:+ShowCodeDetailsInExceptionMessages" "-cp" "C:\Users\RINGGA BUDI UTAMA\AppData\Local\Code\User\workspaceStorage\0a8ce5dd6249c8cb311ce1a7e8e19e3\nedhat.java\jdt_ws\RINGGA-BUDI-UTAMA_sb9621e8\bin" "JOBSHEET12.KubusNoAbsen"

Sisi Kubus: 7
Volume: 343
Luas Permukaan: 294

PS C:\Users\RINGGA BUDI UTAMA\OneDrive\Dokumen\RINGGA-1\Jobsheet7\daspro-jobsheet7\RINGGA-BUDI-UTAMA>

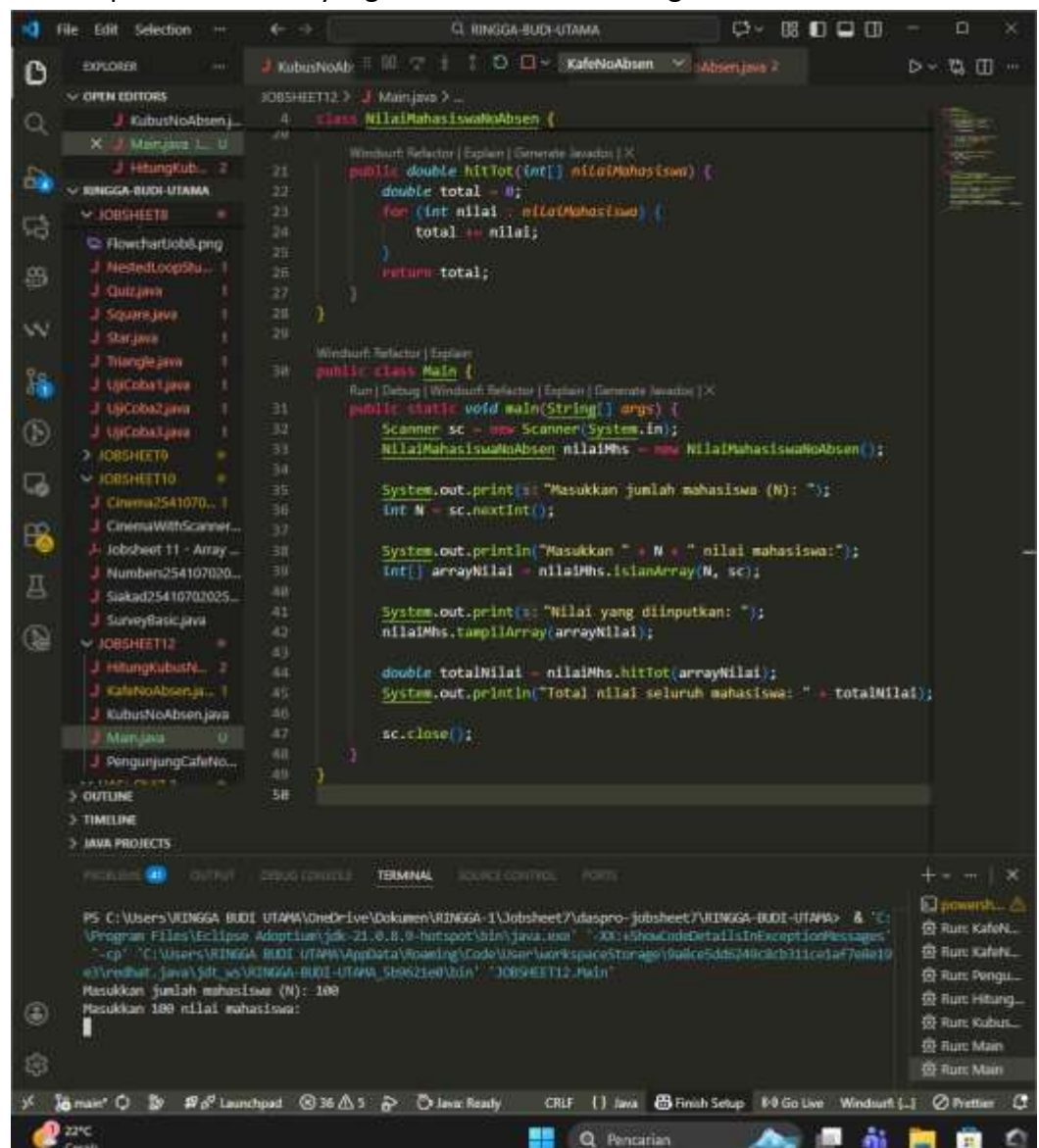
2. Di suatu kelas, terdapat **N** mahasiswa yang mengikuti satu mata kuliah. Nilai tiap mahasiswa untuk mata kuliah tersebut akan diinputkan oleh pengguna dan disimpan dalam sebuah array 1 dimensi berukuran **N**. Buatlah sebuah class **NilaiMahasiswaNoAbsen** yang di dalamnya terdapat beberapa fungsi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



```

class NilaiMahasiswaAbsen {
    public double hitTot(int[] nilaiMahasiswa) {
        double total = 0;
        for (int nilai : nilaiMahasiswa) {
            total += nilai;
        }
        return total;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        NilaiMahasiswaAbsen nilaiMhs = new NilaiMahasiswaAbsen();

        System.out.print("Masukkan jumlah mahasiswa (N): ");
        int N = sc.nextInt();

        System.out.println("Masukkan " + N + " nilai mahasiswa:");
        int[] arrayNilai = nilaiMhs.isianArray(N, sc);

        System.out.print("Nilai yang diinputkan: ");
        nilaiMhs.tampilArray(arrayNilai);

        double totalNilai = nilaiMhs.hitTot(arrayNilai);
        System.out.println("Total nilai seluruh mahasiswa: " + totalNilai);

        sc.close();
    }
}
    
```

Terminal Output:

```

PS C:\Users\UDINGGA_BUDI_UTAMA\OneDrive\Dokumen\UDINGGA-1\Jobsheet\udaspro-jobsheet\UDINGGA-BUDI-UTAMA> & "C:\Program Files\Eclipse Adoptium\jdk-21.0.8-hotspot\bin\java.exe" -Xmx1024m -cp "C:\Users\UDINGGA_BUDI_UTAMA\AppData\Local\Temp\Eclipse\workspace\UDINGGA-BUDI-UTAMA\Jobsheet\udaspro-jobsheet\UDINGGA-BUDI-UTAMA\bin" "Jobsheet12.Main"
Masukkan jumlah mahasiswa (N): 100
Masukkan 100 nilai mahasiswa:
    
```



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 **RekapPenjualanCafeNoAbsen** 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.

- c. 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.



```

package JOBSHEET12;

import java.util.Scanner;

public class RekapPenjualanCafe {

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

        System.out.print("Masukkan jumlah menu: ");
        int menu = sc.nextInt();
        System.out.print("Masukkan jumlah hari: ");
        int hari = sc.nextInt();

        String[] namaMenu = new String[menu];
        int[][] penjualan = new int[menu][hari];

        sc.nextLine(); // buang enter

        for (int i = 0; i < menu; i++) {
            System.out.print("Masukkan nama menu ke-" + (i+1) + ": ");
            namaMenu[i] = sc.nextLine();

            System.out.println("Masukkan penjualan " + namaMenu[i] + " selama");
            for (int j = 0; j < hari; j++) {
                System.out.print("Hari ke-" + (j+1) + ": ");
                penjualan[i][j] = sc.nextInt();
            }
            sc.nextLine();
        }

        // Tampilkan semua data
        System.out.println("\n--- DATA PENJUALAN ---");
        for (int i = 0; i < menu; i++) {
            System.out.print(namaMenu[i] + ": ");
        }
    }
}
    
```

Terminal Output:

```

Hari ke-1: 1

--- DATA PENJUALAN ---
es: 1

Penjualan tertinggi: es (1) pada hari ke-1

--- RATA-RATA PENJUALAN ---
es: 1.0
    
```

Build failed, do you want to continue?

Source: Debugger for Java

Continue Always Continue Fix...

C