

Nama : Nabhila Dita Syakira

NPM : 5520124090

Kelas : IF – C 24

PBO-6

No. 3 Tugas Praktikum PBO

3. Jelaskan perbedaan penggunaan class Scanner dan class BufferedReader dalam Java! Buatlah contoh program yang menunjukkan perbedaan tersebut!

Fitur	Scanner	BufferedReader
Package	java.util	java.io
Kecepatan	Lebih lambat untuk input dalam jumlah besar karena melakukan pemecahan token dan parsing reguler.	Lebih cepat karena menggunakan buffer (penampung memori) untuk membaca data dalam blok besar.
Parsing Data	Otomatis. Memiliki method langsung untuk tipe data primitif seperti nextInt(), nextDouble(), dan nextLine().	Manual. Hanya memiliki method readLine() yang mengembalikan String. Perlu konversi manual (Integer.parseInt(), Double.parseDouble()) untuk mendapatkan tipe data primitif.
Ketergantungan	Berdiri sendiri, langsung membaca dari System.in.	Memerlukan InputStreamReader sebagai jembatan untuk membaca dari stream byte seperti System.in.
Penanganan Error	Tidak memerlukan penanganan IOException secara eksplisit untuk input konsol.	Memerlukan penanganan IOException secara eksplisit (try-catch).
Fungsi Utama	Alat bantu parsing dan pemecah token yang serbaguna.	Alat bantu pembaca stream karakter berkinerja tinggi.

Contoh Program yang Menunjukkan Perbedaan

Kedua program di bawah ini melakukan hal yang sama: meminta input Nama (String) dan Umur (Integer).

1. Contoh Menggunakan Scanner (Parsing Otomatis)

Scanner sangat mudah digunakan karena dapat secara otomatis membaca dan mengonversi tipe data primitif.

```
import java.util.Scanner;

public class ContohScanner {

    public static void main(String[] args) {

        // 1. Inisialisasi: Langsung dari System.in

        Scanner sc = new Scanner(System.in);

        System.out.println("=== Contoh Input dengan Scanner ===");

        System.out.print("Masukkan Nama (String): ");

        String nama = sc.nextLine(); // Membaca String

        System.out.print("Masukkan Umur (int): ");

        // 2. Parsing: Otomatis. Langsung mendapatkan nilai Integer.

        int umur = sc.nextInt();

        System.out.println("\nHasil Scanner:");

        System.out.println("Nama: " + nama);

        System.out.println("Umur: " + umur);

        sc.close();

    }

}
```

2. Contoh Menggunakan BufferedReader (Parsing Manual dan Penanganan IOException)

BufferedReader memerlukan objek perantara (InputStreamReader) dan harus menangani IOException. Semua input dibaca sebagai String, sehingga perlu konversi manual untuk tipe data primitif.

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.io.IOException;

public class ContohBufferedReader {

    public static void main(String[] args) {

        // 1. Inisialisasi: Membutuhkan InputStreamReader sebagai perantara

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("=== Contoh Input dengan BufferedReader ===");

        try {

            System.out.print("Masukkan Nama (String): ");

            String nama = br.readLine(); // Hanya ada method readLine()

            System.out.print("Masukkan Umur (int): ");

            String umurStr = br.readLine(); // Dibaca sebagai String
```

```
// 2. Parsing: Manual. Harus dikonversi dari String ke Integer.
```

```
int umur = Integer.parseInt(umurStr);
```

```
System.out.println("\nHasil BufferedReader:");
```

```
System.out.println("Nama: " + nama);
```

```
System.out.println("Umur: " + umur);
```

```
// 3. Penanganan Error: Wajib menggunakan try-catch untuk IOException
```

```
    } catch (IOException e) {
```

```
        System.err.println("Terjadi error I/O: " + e.getMessage());
```

```
    } catch (NumberFormatException e) {
```

```
        System.err.println("Input harus berupa angka. Terjadi kesalahan format: " + e.getMessage());
```

```
    }
```

```
}
```

```
}
```

Kesimpulan dari Contoh:

- Scanner lebih sederhana untuk input data standar dan konversi tipe data otomatis.
- BufferedReader lebih cepat dan lebih fleksibel untuk operasi I/O yang intensif, tetapi lebih rumit karena membutuhkan InputStreamReader, penanganan IOException, dan konversi tipe data manual.