

LAPORAN SISTEM TERDISTRIBUSI
“TUTORIAL MENGENAI JAVA STREAM”



DISUSUN OLEH
AULIA RAZAK AKMAL
2111082008
TRPL 3B
DOSEN PENGAMPU
ERVAN ASRI.S.Kom.,M.Kom
SEMESTER 5

PRODI D-IV TEKNOLOGI REKAYASA PERANGKAT
LUNAK
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI PADANG
2023/2024

Tutorial Java Stream

Pengenalan Java Stream

Java Stream adalah sebuah fitur yang diperkenalkan dalam Java 8 untuk memungkinkan pemrograman fungsional dengan koleksi data. Stream adalah urutan elemen yang dapat diolah secara sekuensial atau paralel. Dengan Java Stream, Anda dapat melakukan operasi seperti filter, map, dan reduce pada koleksi data dengan lebih mudah dan efisien.

Mengimpor Paket

Sebelum Anda dapat menggunakan Java Stream, pastikan untuk mengimpor paket yang diperlukan:

```
J javaStream.java
1  import java.util.stream.*;
2  import java.util.List;
3
```

Membuat Stream

Anda dapat membuat stream dari berbagai sumber data seperti koleksi, array, atau bahkan dari elemen-elemen individu.

Membuat Stream dari List

```
// //list
List<String> daftarNama = Arrays.asList("Ananta", "Budi", "Coki", "Damar");
// // Membuat Stream dari List
Stream<String> streamNama = daftarNama.stream();
```

Membuat Stream dari Array

```
// //array
int[] angka = {1, 2, 3, 4, 5, 6, 7};
// // Membuat Stream dari Array
IntStream streamAngka = Arrays.stream(angka);
```

Operasi Stream

Berikut adalah beberapa operasi yang dapat Anda lakukan dengan Java Stream:

1. Filter

Menggunakan operasi **filter** untuk memfilter elemen berdasarkan kondisi tertentu.

```
// //FILTER
Stream<String> streamNama = daftarNama.stream();
// // Menyaring nama yang panjangnya lebih dari 4 karakter
Stream<String> namaPanjang = streamNama.filter(nama -> nama.length() > 4);
```

2. Map

Menggunakan operasi **map** untuk mentransformasi elemen-elemen stream.

```
// //MAP
Stream<String> streamNama = daftarNama.stream();
// // Mengubah semua nama menjadi huruf besar
Stream<String> namaBesar = streamNama.map(nama -> nama.toUpperCase());
```

3. Reduce

Menggunakan operasi **reduce** untuk menggabungkan elemen-elemen stream menjadi satu hasil.

```
// //REDUCE
Stream<Integer> streamAngka = Stream.of(1, 2, 3, 4, 5, 6, 7);
// // Menghitung jumlah semua angka dalam stream
int jumlah = streamAngka.reduce(0, (a, b) -> a + b);
```

Contoh Program :

1. Menghitung Jumlah Bilangan Genap

```
D: > Folder Razak > KULIAH > SEMESTER 5 > SISTER > J stream.java > stream > main(String[])

1  import java.util.stream.*;
2  import java.util.Arrays;
3  import java.util.List;
4
5  public class stream{
6      Run | Debug
7      public static void main(String[] args) {
8          List<Integer> angka = Arrays.asList(...a:1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
9
10         int jumlahGenap = angka.stream()
11             .filter(x -> x % 2 == 0)
12             .mapToInt(Integer::intValue)
13             .sum();
14         System.out.println("Jumlah bilangan genap: " + jumlahGenap);
15     }
16 }

PS C:\Users\Zikra> & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Zikra\AppData\Local\Temp\vscode\ws_c5696\jdt_ws\jdt.ls-java-project\bin' 'stream'
Jumlah bilangan genap: 30
PS C:\Users\Zikra>
```

2. Menemukan Nama yang Dimulai dengan huruf “B”

```
public class stream {
    Run | Debug
    public static void main(String[] args) {
        List<String> daftarNama = Arrays.asList(...a:"Ananta", "Ananda", "Budi", "Damar");
        List<String> namaAwalB = daftarNama.stream()
            .filter(nama -> nama.startsWith(prefix:"B"))
            .collect(Collectors.toList());

        System.out.println("Nama yang dimulai dengan 'B': " + namaAwalB);
    }
}

PS C:\Users\Zikra> & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Zikra\AppData\Local\Temp\vscode\ws_c5696\jdt_ws\jdt.ls-java-project\bin' 'stream'
Nama yang dimulai dengan 'B': [Budi]
PS C:\Users\Zikra>
```