

LAPORAN SISTEM TERDISTRIBUSI

“TUTORIAL JAVA STREAM”



OLEH :

FAZILA SURYA AZZAHRAH - 2111081005

3A TRPL

DOSEN PENGAMPU :

ERVAN ASRI, S.Kom., M.Kom

SEMESTER 5

PRODI TEKNOLOGI REKAYASA PERANGKAT LUNAK

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI PADANG

1. Apa Itu Java Stream ?

Java Stream diperkenalkan pertama kali di Java versi 8 atau 1.8, java stream adalah implementasi di java untuk aliran data. Stream adalah rangkaian objek yang mendukung berbagai metode yang dapat disalurkan untuk menghasilkan hasil yang diinginkan.

Berbeda dengan Array atau Collection yang berfokus terhadap data, Stream berfokus pada operasi apa yang dilakukan. Aliran tersebut tidak mengubah struktur data asli, melainkan hanya memberikan hasil sesuai metode yang disalurkan.

Untuk membuat Stream di java ada banyak cara, bisa membuat langsung Stream dari Array, atau dari Collection. Ada beberapa operasi berbeda pada Stream :

1) Intermediate Operations

- a. Map : mengembalikan aliran yang terdiri dari hasil penerapan fungsi yang diberikan ke elemen aliran
- b. Filter : memilih elemen sesuai predikat yang diteruskan sebagai argument
- c. Sorted : digunakan untuk mengurutkan aliran

2) Terminal Operations

- a. Collect : mengembalikan hasil operasi perantara yang dilakukan pada aliran
- b. forEach : melakukan iterasi melalui setiap elemen aliran
- c. Reduce : mengurangi elemen aliran menjadi satu nilai. Metode pengurangan menggunakan Binary Operator sebagai parameter

2. Contoh Program Java Stream

1) Menggunakan Stream dengan operasi filter (Retrieving Operation)

Retrieving operation dalam Java Stream digunakan untuk mengambil elemen-elemen tertentu dari stream yang memenuhi suatu kondisi tertentu, contoh retrieving operationnya adalah operasi '**filter**'

```

1  import java.util.Arrays;
2  import java.util.List;
3  import java.util.stream.Collectors;
4  /**
5   *
6   * @author Asus
7   */
8  public class RetrievingOperation {
9      public static void main(String[] args) {
10         List<String> words = Arrays.asList("purple", "blue", "black",
11         "yellow", "pink", "magenta");
12         // Menggunakan Stream dan operasi filter untuk mengambil kata-
13         kata yang panjangnya lebih dari 4 karakter
14         List<String> filteredWords = words.stream()
15             .filter(word -> word.length() > 4)
16             .collect(Collectors.toList());
17         System.out.println("Kata-kata yang panjangnya lebih dari 4 kar
18         akter: " + filteredWords);
19     }
20 }

```

Output :

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL Run: RetrievingOperation
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\USER> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInE
xceptionMessages' '-cp' 'C:\Users\USER\AppData\Local\Temp\vscodesws_1bc98\jdt_ws\jdt.ls-j
ava-project\bin' 'RetrievingOperation'
Kata-kata yang panjangnya lebih dari 4 karakter: [purple, black, yellow, magenta]
PS C:\Users\USER>

```

Penjelasan :

Pada contoh diatas, pertama kita membuat sebuah list '**words**' yang berisi beberapa kata. Kemudian kita menggunakan Java Stream untuk mengambil kata-kata yang memiliki Panjang karakter lebih dari 5 dengan menggunakan operasi '**.filter(word -> word.length() > 4)**'. Setelah itu hasil dari aliran data tersebut dikumpulkan Kembali menjadi List dengan '**.collect(Collectors.toList())**'.

2) Menggunakan stream dan operasi map (Map Operation)

Map operation digunakan untuk mengembalikan Stream yang terdiri dari penerapan fungsi yang diberikan ke elemen-elemen Stream.

```
1 import java.util.Arrays;
2 import java.util.List;
3 import java.util.stream.Collectors;
4
5 public class MapOperation {
6     public static void main(String[] args) {
7         List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);
8
9         // Menggandakan stream dan operasi map untuk menggandakan
10        // an nilai pada list array
11        List<Integer> doubledNumbers = numbers.stream()
12            .map(n -> n * 2)
13            .collect(Collectors.toList());
14
15        System.out.println("Angka asli: " + numbers);
16        System.out.println("Hasil Angka yang digandakan: " + do
17        ubledNumbers);
18    }
19 }
```

Output :

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

Open folder in new window (ctrl + click) :files\Java\jdk-19\bin\java.exe' '-XX:+Show
CodeDetailsInExceptionMessages' '-cp' 'C:\Users\USER\AppData\Local\Temp\vsc
odesws_1bc98\jdt_ws\jdt.ls-java-project\bin' 'MapOperation'
Angka asli: [1, 2, 3, 4, 5]
Hasil Angka yang digandakan: [2, 4, 6, 8, 10]
PS C:\Users\USER>
```

Penjelasan :

Pada contoh ke-2 ini pertama kita juga membuat List '**numbers**' yang berisi beberapa angka. Kemudian kita menggunakan Java Stream untuk menggandakan

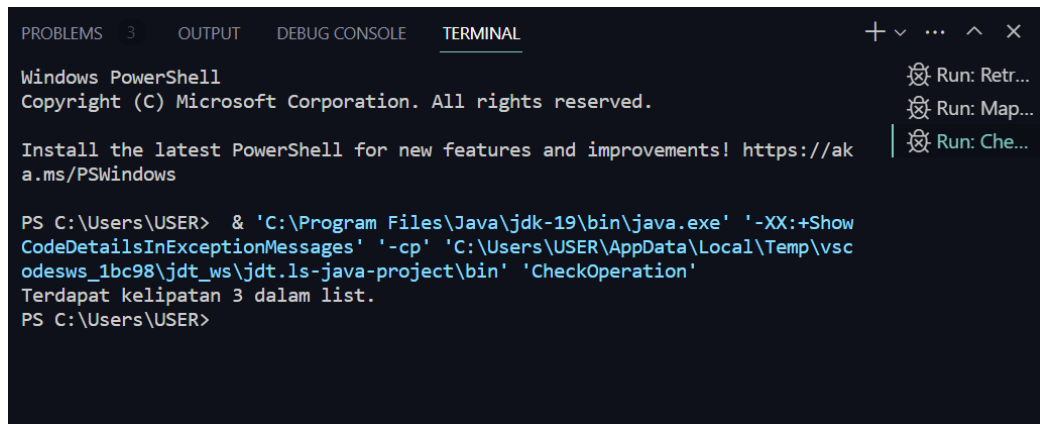
setiap angka dalam list tersebut dengan menggunakan operasi `\.map(n -> n * 2)'` untuk menggandakan (kali 2) setiap angka. Setelah itu hasilnya dikumpulkan Kembali menjadi List dengan `\.collect(Collectors.toList())'`.

3) Menggunakan Stream dan operasi anyMatch (Check Operation)

Check operation digunakan untuk memeriksa apakah elemen-elemen dalam stream memenuhi suatu kondisi tertentu, contoh check operation yang sederhana menggunakan operasi `'anyMatch'`

```
1  import java.util.Arrays;
2  import java.util.List;
3  /**
4   *
5   * @author Asus
6   */
7  public class CheckOperation {
8      public static void main(String[] args) {
9          List<Integer> numbers = Arrays.asList(3, 5, 6, 8, 10, 1
10         2);
11          // Menggunakan Stream dan operasi anyMatch untuk memeri
12          ksa apakah ada angka kelipatan 3 dalam list
13          boolean hasEvenNumber = numbers.stream()
14              .anyMatch(n -> n % 3 == 0);
15          if (hasEvenNumber) {
16              System.out.println("Terdapat kelipatan 3 dalam lis
17          t.");
18          } else {
19              System.out.println("Tidak ada kelipatan 3 dalam lis
20          t.");
21          }
22      }
23  }
```

Output :



```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\USER> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\USER\AppData\Local\Temp\vscodesws_1bc98\jdt_ws\jdt.ls-java-project\bin' 'CheckOperation'
Terdapat kelipatan 3 dalam list.
PS C:\Users\USER>
```

Penjelasan :

Pada contoh ke-3 ini pertama kita juga membuat List `'numbers'` yang berisi beberapa angka. Kemudian dengan menggunakan Java Stream dan operasi `'.anyMatch(n -> n % 3 == 0)'` untuk memeriksa apakah ada angka kelipatan 3 dalam list. Kondisi `'n % 3 == 0'` digunakan untuk memeriksa apakah angka tersebut kelipatan 3. Setelah itu hasilnya dikumpulkan Kembali menjadi List dengan `'.collect(Collectors.toList())'`.