SISTER TERDISTRIBUSI

Tutorial Java Stream



Disusun Oleh :

Nama : Bulan Fitria Adviani

Kelas : TRPL 3B

No BP : 2111082009

JURUSAN TEKNOLOGI INFORMASI

PROGRAM STUDI TRPL

POLITEKNIK NEGERI PADANG

2023

Java stream adalah fitul yang diperkenalkan dalam java 8 untuk memungkinkan pemrosesan data berbasis koleksi dengan lebih mudah, fungsional dan deklaratif. Stream memungkinkan melakukan berbagai operasi pada data koneksi seperti list, set, map, array atau bahkan data dari sumber lain dengan pendekatan yang mirip dengan SQL atau pemrograman fungsional.

Contoh :

1. Filtering Data

Pemrograman dan pengelolahan data, merujuk pada proses pengambilan atau penyelesian elemen tertentu dari sekumpulan data berdasarkan kriteria tertentu.

*import java.util.Arrays;*

*import java.util.List;*

*import java.util.stream.Collectors;*

*public class StreamExample {*

*public static void main(String[] args) {*

*List<String> fruits = Arrays.asList("apple", "banana", "cherry", "date", "elderberry");*

*List<String> filteredFruits = fruits.stream()*

*.filter(fruit -> fruit.startsWith("b"))*

*.collect(Collectors.toList());*

*System.out.println(filteredFruits);*

*}*

*}*

1. Reducing Data

Pemrograman dan pengolahan data, merujuk pada proses mengagregasi atau menggabungkan sekumpulan data menjadi satu nilai tunggal berdasarkan operasi tertentu.

*import java.util.Arrays;*

*import java.util.List;*

*import java.util.Optional;*

*public class StreamExample {*

*public static void main(String[] args) {*

*List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);*

*Optional<Integer> sum = numbers.stream()*

*.reduce((a, b) -> a + b);*

*sum.ifPresent(s -> System.out.println("Sum: " + s));*

*}*

*}*

1. Maaping Data

Pemrograman dan pengolahan data, merujuk pada proses mengubah atau mengonversi data dari satu bentuk atau format ke bentuk format lainnya.

*import java.util.Arrays;*

*import java.util.List;*

*import java.util.stream.Collectors;*

*public class StreamExample {*

*public static void main(String[] args) {*

*List<String> fruits = Arrays.asList("apple", "banana", "cherry");*

*List<Integer> fruitLengths = fruits.stream()*

*.map(String::length)*

*.collect(Collectors.toList());*

*System.out.println(fruitLengths);*

*}*

*}*