**SISTEM TERDISTRIBUSI**

**“Tutorial Java Stream”**



**Disusun Oleh :**

**Nadia Monika Putri**

**2111082034**

**3B**

**Teknologi Rekayasa Perangkat Lunak**

**Teknologi Informasi**

**Politeknik Negeri Padang**

**2023**

Java Stream adalah sebuah fitur baru yang diperkenalkan pada Java 8 untuk memproses koleksi objek secara deklaratif dan fungsional. Stream pada Java 8 adalah sebuah layer abstrak yang memungkinkan pengguna untuk memproses data dengan cara yang mirip dengan pernyataan SQL. Stream terdiri dari sumber data yang diikuti oleh satu atau lebih metode intermediate yang digabungkan bersama (pipelined) dan sebuah metode terminal untuk memproses hasil akhir. Stream pada Java 8 tidak sama dengan struktur data koleksi, melainkan mengambil input dari koleksi, array, atau channel I/O (seperti file).

Stream pada Java 8 memiliki beberapa fitur, antara lain :

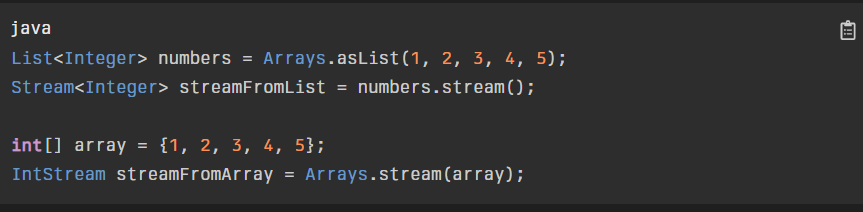
* Stream tidak mengubah struktur data asli, melainkan hanya memberikan hasil sesuai dengan metode pipelined yang digunakan.
* Stream dapat digunakan untuk memproses koleksi objek secara parallel.
* Stream dapat digunakan untuk memproses tipe data primitif seperti int, long, dan double.

Dalam penggunaannya, Stream pada Java 8 dapat digunakan untuk melakukan berbagai macam operasi pada koleksi objek, seperti filter, map, reduce, dan lain sebagainya. Stream pada Java 8 juga dapat digunakan untuk menggantikan konstruksi yang verbose seperti while dan for dalam pemrosesan koleksi objek.

Berikut adalah tutorial singkat mengenai Java Stream :

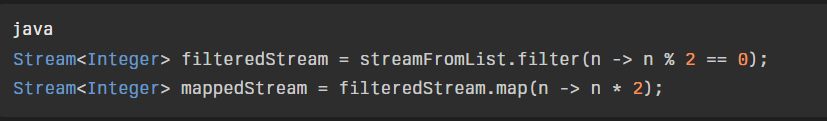
1. Mengambil Stream dari Koleksi atau Array

* Untuk mengambil stream dari koleksi, panggil method stream() pada objek koleksi tersebut.
* Untuk mengambil stream dari array, panggil method Arrays.stream(array).



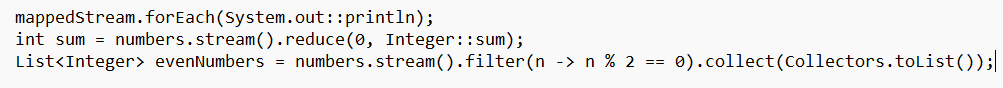
1. Intermediate Operations

* Intermediate operations adalah operasi yang dilakukan pada stream sebelum operasi terminal.
* Beberapa intermediate operations yang umum digunakan adalah filter(), map(), flatMap(), distinct(), sorted(), dan limit().



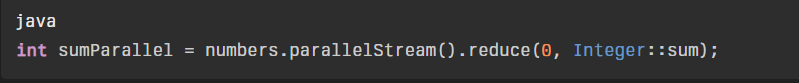
1. Terminal Operations

* Terminal operations adalah operasi yang dilakukan pada stream untuk menghasilkan output.
* Beberapa terminal operations yang umum digunakan adalah forEach(), collect(), reduce(), count(), dan anyMatch().



1. Parallel Stream

* Parallel stream memungkinkan penggunaan beberapa thread untuk memproses stream secara parallel.
* Untuk mengubah stream menjadi parallel stream, panggil method parallel() pada stream tersebut.



Contoh penggunaan Java Stream :

