SISTEM TERDISTRIBUSI

DISUSUN OLEH:

DINDA GATYA RABBANI

2111082012

TRPL 3B

DOSEN PENGAMPU :

ERVAN ASRI, S.Kom., M.Kom

SEMESTER V

JURUSAN TEKNOLOGI INFORMASI

PROGRAM STUDI D4 TEKNOLOGI REKAYASA PERANGKAT LUNAK

POLITEKNIK NEGERI PADANG

Tutorial Java Stream

Java Stream API diperkenalkan pada rilis JDK 8 dengan fungsi utama untuk mempermudah pemrosesan elemen-elemen data secara kolektif menggunakan operasi-operasi fungsional. Java Stream API disediakan pada paket java.util.stream. Selain bisa digunakan secara independen, API ini juga bisa diakses melalui Java Collection API.

Dalam *stream* API, terdapat 4 *interface* inti yaitu:
IntStream, LongStream, DoubleStream dan Stream<T>. Sesuai namanya, masingmasing *stream* digunakan untuk elemen dengan tipe Integer, Long, dan Double terkecuali *interface* Stream<T>.

Ada beberapa operasi pada Java Stream :

Operasi Menengah

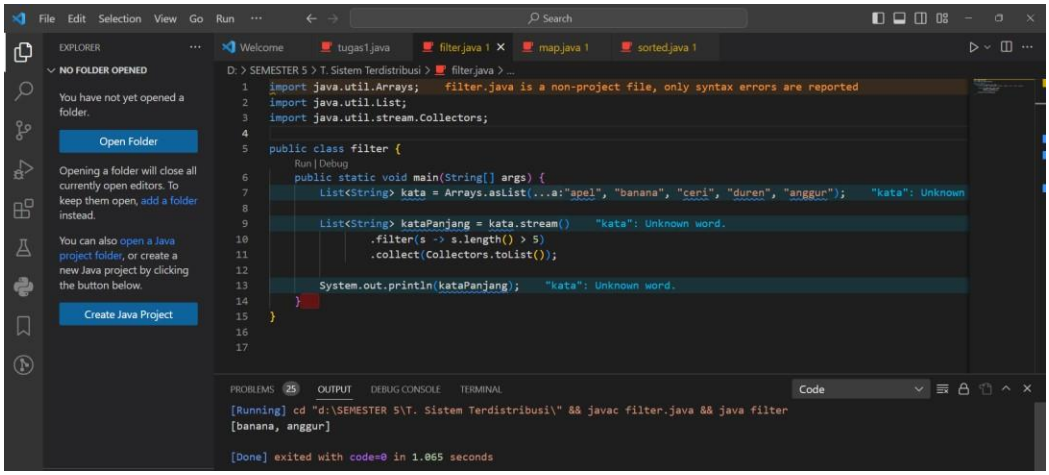
- map: Metode map digunakan untuk mengembalikan aliran yang terdiri dari hasil penerapan fungsi yang diberikan ke elemen aliran ini.
- filter: Metode filter digunakan untuk memilih elemen sesuai Predikat yang diteruskan sebagai argumen.
- sorted: Metode yang diurutkan digunakan untuk mengurutkan aliran.

Operasi Terminal

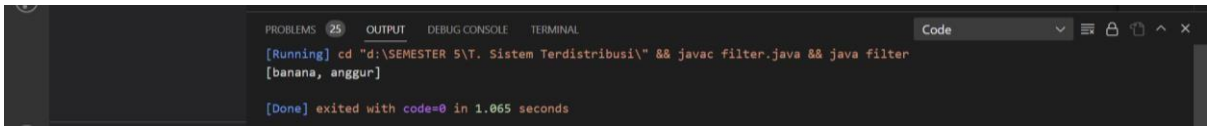
- kumpulkan: Metode pengumpulan digunakan untuk mengembalikan hasil operasi perantara yang dilakukan pada aliran.
- forEach: Metode forEach digunakan untuk melakukan iterasi melalui setiap elemen aliran.
- pengurangan: Metode pengurangan digunakan untuk mengurangi elemen aliran menjadi satu nilai. Metode pengurangan menggunakan BinaryOperator sebagai parameter.

Dan berikut beberapa Contoh program pada operasi Java Stream :

1. Menggunakan operasi filter



Output :



Penjelasan :

Kode di atas membuat sebuah list **kata** yang berisi beberapa kata.

Kemudian, kita menggunakan Stream pada list **kata** dengan **kata.stream()**.

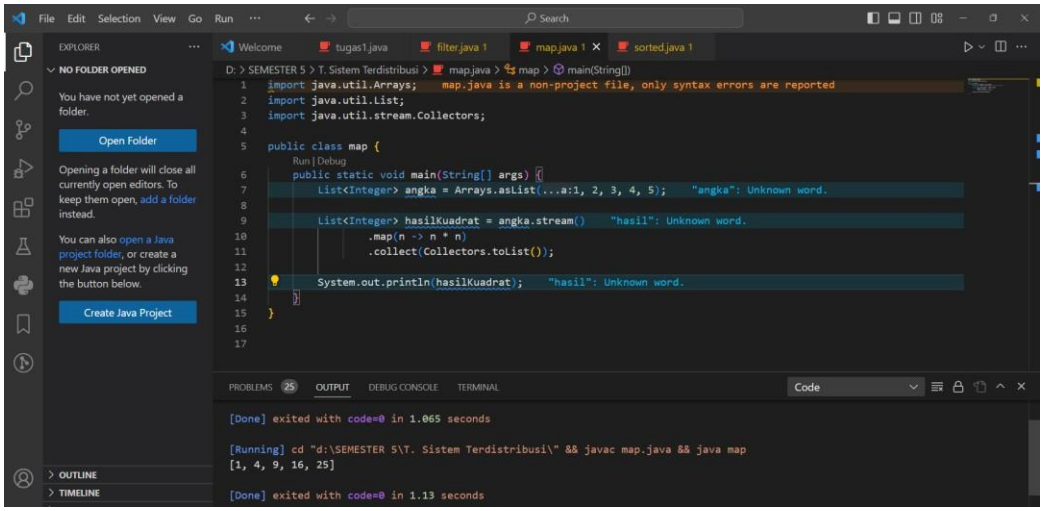
Pada Stream tersebut, kita menggunakan operasi **filter**. **.filter()** digunakan untuk menyaring elemen-elemen yang memenuhi suatu kondisi tertentu. Dalam contoh ini, kondisinya adalah panjang string lebih dari 5 karakter, yang diwakili oleh ekspresi lambda **s -> s.length() > 5**.

Hasil dari operasi **filter** ini adalah sebuah Stream baru yang hanya berisi kata-kata yang panjangnya lebih dari 5 huruf.

Akhirnya, kita menggunakan **.collect(Collectors.toList())** untuk mengumpulkan hasilnya ke dalam List baru **kataPanjang**.

Hasilnya adalah List baru yang hanya berisi kata "banana" karena kata ini adalah satusatunya yang memiliki panjang lebih dari 5 huruf.

2. Menggunakan Operasi Map



Output :

Penjelasan :

Kode di atas membuat sebuah list **angka** yang berisi beberapa angka.

Kemudian, kita menggunakan Stream pada list **angka** dengan **angka.stream()**.

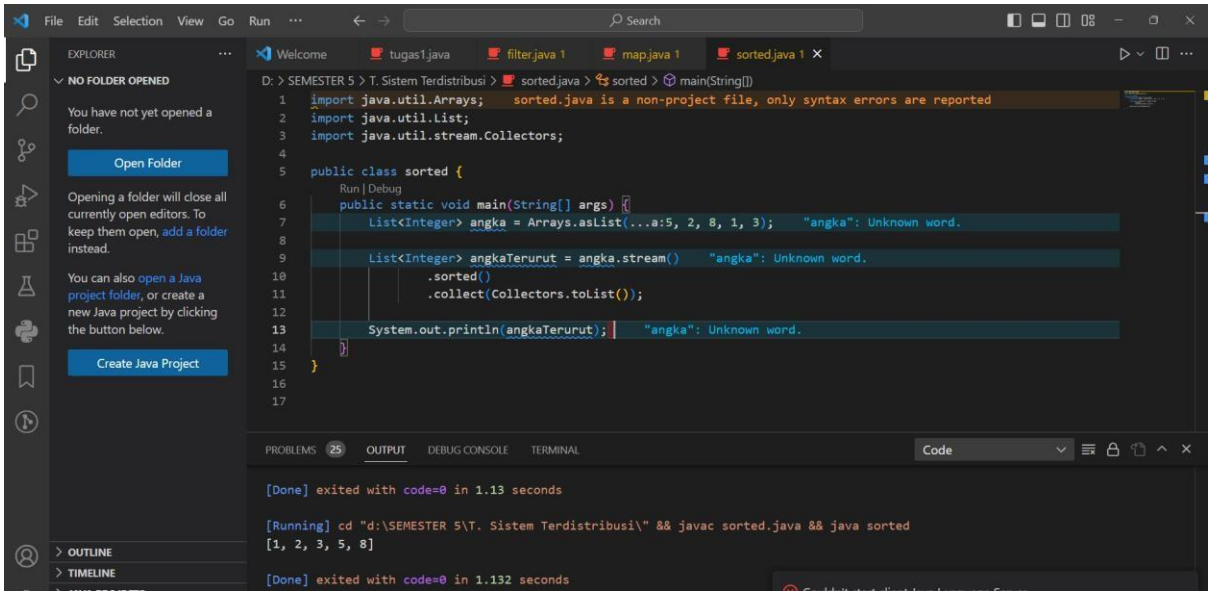
Pada Stream tersebut, kita menggunakan operasi **map**. **.map()** digunakan untuk mengubah setiap elemen dalam Stream sesuai dengan ekspresi lambda yang diberikan. Dalam contoh ini, kita mengkuadratkan setiap angka dengan ekspresi **n -> n * n**.

Hasil dari operasi **map** ini adalah Stream baru yang berisi hasil pengkuadratan setiap angka dalam list.

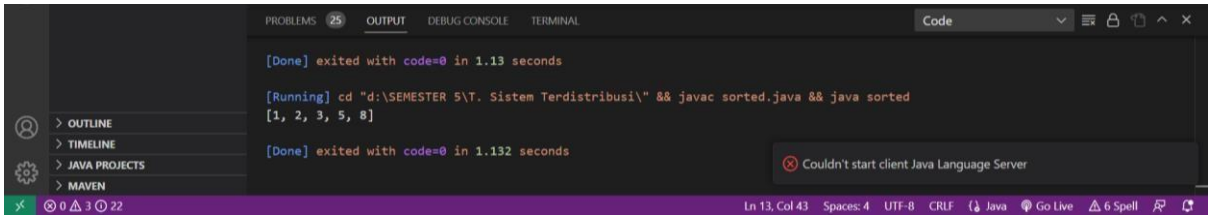
Akhirnya, kita menggunakan **.collect(Collectors.toList())** untuk mengumpulkan hasilnya ke dalam List baru **hasilKuadrat**.

Hasilnya adalah List baru yang berisi hasil kuadrat dari setiap angka dalam list asalnya.

3. Menggunakan Operasi Sorted



Output:



Penjelasan :

Kode di atas membuat sebuah list **angka** yang berisi beberapa angka yang tidak terurut.

Kemudian, kita menggunakan Stream pada list **angka** dengan **angka.stream()**.

Pada Stream tersebut, kita menggunakan operasi **sorted**. **.sorted()** digunakan untuk mengurutkan elemen-elemen dalam Stream. Dalam contoh ini, kita tidak memberikan kriteria pengurutan khusus, sehingga angka-angka akan diurutkan secara ascending (menaik) secara default.

Hasil dari operasi **sorted** ini adalah Stream baru yang berisi angka-angka yang sudah terurut.

Akhirnya, kita menggunakan **.collect(Collectors.toList())** untuk mengumpulkan hasilnya ke dalam List baru **angkaTerurut**.

Hasilnya adalah List baru yang berisi angka-angka yang sudah diurutkan dari terkecil ke terbesar.