

JAVA STREAM

Java Stream adalah fitur yang diperkenalkan dalam Java 8 untuk memudahkan pemrosesan data dalam bentuk koleksi atau array. Stream memungkinkan pengguna untuk menggambarkan operasi pada data sebagai ekspresi fungsional, yang dapat dijalankan secara sekuensial atau paralel.

Dasar-Dasar Java Stream

Stream vs. Collection

- Stream adalah sekumpulan elemen yang dihasilkan secara fungsional dari sumber seperti koleksi, array, atau bahkan I/O channel.
- Collection, di sisi lain, adalah struktur data yang menyimpan dan mengelola elemen secara eksplisit.

Mendapatkan Stream dari Koleksi atau Array

```
List<String> list = Arrays.asList("a", "b", "c");  
Stream<String> streamFromList = list.stream();  
  
int[] array = {1, 2, 3, 4, 5};  
IntStream streamFromArray = Arrays.stream(array);
```

Mapping

- Mengubah setiap elemen dengan fungsi tertentu.

Contoh map

```
List<Integer> lengths = list.stream()  
    .map(String::length)  
    .collect(Collectors.toList());
```

Sorting

- Mengurutkan elemen berdasarkan suatu kriteria.

Contoh sorted

```
List<String> sortedList = list.stream()  
    .sorted()  
    .collect(Collectors.toList());
```

Kelebihan dan Kelemahan Java Stream

Kelebihan

- Kode yang lebih ringkas dan ekspresif.
- Mendukung pengolahan paralel untuk kinerja yang lebih baik.

Kelemahan

- Memerlukan pemahaman yang mendalam.
- Tidak cocok untuk semua jenis operasi.

. Kesimpulan

Java Stream memberikan pendekatan fungsional dan deklaratif untuk memproses data, membuat kode lebih bersih dan mudah dibaca. Meskipun ada kekurangan, manfaatnya dalam pengembangan aplikasi modern sangat signifikan. Dengan memahami konsep-konsep dasar dan contoh penggunaan, pengembang dapat memanfaatkan Java Stream secara efektif dalam proyek mereka.