

JAVA STREAM

Java Streams adalah fitur yang sangat kuat yang diperkenalkan dalam Java 8 untuk memproses koleksi data secara efisien. Mereka memungkinkan Anda untuk melakukan berbagai operasi pada data koleksi seperti filtering, mapping, sorting, dan reduksi dengan sintaks yang elegan. Dalam tutorial ini, kita akan membahas beberapa contoh program Java menggunakan Java Streams.

Contoh 1: Filtering dengan Java Stream

```
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class StreamExample1 {
    public static void main(String[] args) {
        List<String> words = Arrays.asList("apple", "banana", "cherry", "date", "elderberry");

        List<String> filteredWords = words.stream()
            .filter(word -> word.startsWith("a"))
            .collect(Collectors.toList());

        System.out.println(filteredWords);
    }
}
```

Contoh 2: Mapping dengan Java Stream

```
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class StreamExample2 {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("John", "Alice", "Bob", "Eve");

        List<String> uppercaseNames = names.stream()
            .map(String::toUpperCase)
            .collect(Collectors.toList());

        System.out.println(uppercaseNames);
    }
}
```

Contoh 3: Sorting dengan Java Stream

```
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class StreamExample3 {
    public static void main(String[] args) {
        List<Integer> numbers = Arrays.asList(5, 2, 8, 1, 9);

        List<Integer> sortedNumbers = numbers.stream()
            .sorted()
            .collect(Collectors.toList());

        System.out.println(sortedNumbers);
    }
}
```

Contoh 4: Reduksi dengan Java Stream

```
import java.util.Arrays;
import java.util.List;
import java.util.Optional;

public class StreamExample4 {
    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);

        if (sum.isPresent()) {
            System.out.println("Jumlah: " + sum.get());
        }
    }
}
```