**JAVA STREAM**

## **Pendahuluan**

## ## Sebelum Belajar Materi ini ##

## sebaiknya mempelajari Java Dasar, Java Object Oriented Programming, Java Generic, Java Collection, Java Lambda, Apache Maven, Java Unit Test

## ## Agenda ##

Pengenalan Java Stream, Membuat Stream, Lazy Evaluation, Stream Builder, Stream Operations, Collectors, Parallel Stream

## **Pengenalan Java Stream**

* Java Stream diperkenalkan pertama kali di Java versi 8 atau 1.8
* Java Stream adalah implementasi di Java untuk aliran data
* Java Stream erat kaitannya dengan Java Collection dan Lambda Expression
* Semua fitur Java Stream terdapat di package java.util.stream
* <https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/util/stream/package-summary.html>

## Apa itu Stream? ##

* Stream adalah aliran data
* Berbeda dengan Array atau Collection yang merupakan kumpulan data
* Biasanya dalam Array atau Collection, kita selalu focus terhadap data, sedangkan dalam Stream, kita akan focus terhadap operasi apa yang akan kita lakukan

## Class java.util.stream.Stream<T> ##

* Java Stream diimplementasikan oleh sebuah class bernama Stream di dalam package java.util.stream
* Java Stream tidak sekompleks Java Collection, namun walaupun sederhana, fitur Java Stream sangat powerfull
* Class Stream merupakan Generic Class, sehingga kitab isa membuat aliran data dengan tipe yang kita inginkan

## **Membuat Stream**

* Ada banyak cara untuk membuat Stream di Java
* Bisa membuat langsung Stream dari Array, atau dari Collection

## Menjalankan Stream ##

* Secara default, Stream itu bersifat Cold, artinya data di Stream tidak akan mengalir sampai kita memintanya
* Ada banyak cara untuk meminta Stream mulai mengalirkan datanya, hal ini akan kita bahas di bagian Stream Operations
* Untuk saat ini, salah satu method yang bis akita gunakan agar aliran data di Stream mengalir adalah menggunakan method forEach
* Stream hanya bisa jalan sekali, mirip seperti aliran data, setelah mengalir, aliran data tidak bisa diulang lagi dari awal

## **Stream Builder**

* Kadang kita ingin membuat Stream secara manual, seperti menambahkan datanya ke Stream secara manual misalnya
* Java menyediakan Stream Builder untuk membuat Stream secara manual, dan kita bisa menggunakannya seperti Collection, kita bisa membuat Stream Builder, dan menambahkan data ke Stream Builder, setelah selesai, baru kita buat Stream nya
* https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/util/stream/Stream.Builder.html

## **Stream Operations**

* Berbeda dengan Java Collection, di Java Stream hal yang sangat menarik adalah fitur Stream Operations nya
* Stream Operations adalah kumpulan operasi-operasi yang bis akita gunakan untuk memanipulasi Stream itu sendiri
* Ada baanyak sekali jenis – jenis Stream Operation yang nanti akan kita bahas satu persatu
* Namun secara garis besar, Stream Operations tidak akan memodifikasi data aslinya, melainkan hasil dari Stream Operation adalah sebuah Stream baru

## **Stream Pipiline**

* Saat kita membuat Stream, biasanya kita akan melakukan banyak operasi terhadap Stream tersebut
* Dan biasanya kita akan membuat Stream Pipiline
* Stream Pipiline terdiri dari sebuah suber stream(bisa array, collection dan lain-lain), lalu diikuti dengan kosong atau lebih stream operations dan diakhiri dengan operasi akhir, misalnya forEach
* Saat menggunakan Stream, hamper kebanyakan kita pasti akan membuat sebuah Stream Pipiline