

---

# Pengenalan Virtual Thread



# Pengenalan Virtual Thread

- Virtual Thread adalah salah satu fitur baru yang ada Java 21
- Virtual Thread diajukan menggunakan proposal JEP 444
- <https://openjdk.org/jeps/444>
- Virtual Thread adalah thread ringan yang secara dramatis meringankan beban ketika membuat aplikasi yang membutuhkan concurrency yang tinggi



## Perbedaan dengan Thread

- Thread di Java sebenarnya adalah wrapper Thread di Sistem Operasi
- Artinya pembuatan Thread di Java sama mahalanya dengan Thread di OS
- Satu thread biasanya bisa memakan konsumsi memory sebesar 1-2MB, sehingga kita tidak bisa membuat terlalu banyak Thread, misal sampai jutaan, karena bisa mendapat masalah kehabisan memory
- Oleh karena keterbatasan ini, Virtual Thread diajukan
- Virtual Thread konsepnya sama seperti Goroutine di Golang atau Coroutine di Kotlin



# Cara Kerja Virtual Thread

- Virtual Thread sendiri tetap akan berjalan diatas Thread, namun yang membedakan adalah, kita tidak butuh membuat Thread secara manual, karena itu dikelola oleh JVM
- Virtual Thread akan dijalankan dalam ForkJoinPool Thread, sehingga Virtual Thread akan berjalan diatas Thread yang berjumlah total seperti jumlah CPU Core yang kita gunakan
- Karena Virtual Thread sangat ringan, oleh karena itu kita bisa buat sampai jutaan Virtual Thread secara mudah tanpa harus takut kehabisan memory

---

# Membuat Project



# Membuat Project

- <https://start.spring.io/>

---

# Platform Thread



# Platform Thread

- Sekarang, Thread biasa yang sebelumnya kita gunakan, disebut dengan nama Platform Thread
- Ketika belajar Thread pada kelas Java Thread, semua itu kita sebut adalah Platform Thread
- Hal ini dilakukan agar kita tidak bingung dengan Virtual Thread
- Platform Thread merupakan wrapper untuk Thread Sistem Operasi, artinya setiap Platform Thread yang dibuat, secara otomatis akan dibuat Thread Sistem Operasi
- Platform Thread bisa digunakan untuk jenis pekerjaan apapun, namun limitasinya hanya pada resource yang digunakan





# Membuat Platform Thread

- Untuk membuat Platform Thread, kita bisa gunakan cara seperti pada Kelas Java Thread
- Namun di Java 21, terdapat sebuah Builder Platform Thread yang bisa digunakan untuk mempermudah ketika kita membuat Platform Thread, yaitu menggunakan method `Thread.ofPlatform()`



## Kode : Platform Thread

```
public class PlatformThreadApp {  
  
    public static void main(String[] args) {  
        for (int i = 0; i < 10; i++) {  
            Thread.ofPlatform().name("eko").daemon(false).start(() → {  
                try {  
                    Thread.sleep(Duration.ofSeconds(2));  
                    System.out.println("Hello " + Thread.currentThread() + "!");  
                } catch (InterruptedException e) {  
                    throw new RuntimeException(e);  
                }  
            });  
        }  
    }  
}
```

---

# Virtual Thread



# Virtual Thread

- Cara membuat Virtual Thread adalah dengan menggunakan `Builder Thread.ofVirtual()`
- Implementasi dari Virtual Thread sebenarnya sama saja, yaitu class `java.lang.Thread`, jadi fitur yang bisa digunakan di Platform Thread bisa digunakan juga di Virtual Thread
- Perbedaan bisa terlihat pada detail Thread ketika kita jalankan



## Code : Virtual Thread

```
public class VirtualThreadApp {  
  
    public static void main(String[] args) throws IOException {  
        for (int i = 0; i < 10000; i++) {  
            Thread thread = Thread.ofVirtual().name("eko").unstarted(() -> {  
                try {  
                    Thread.sleep(Duration.ofSeconds(2));  
                    System.out.println("Hello " + Thread.currentThread() + "!");  
                } catch (InterruptedException e) {  
                    throw new RuntimeException(e);  
                }  
            });  
            thread.start();  
        }  
  
        System.in.read();  
    }  
}
```

---

# Virtual Thread Executor Service



# Virtual Thread Executor Service

- Saat menggunakan Thread, kita biasanya jarang menggunakan Thread secara manual
- Biasanya kita akan menggunakan Executor Service
- Virtual Thread juga memiliki Executor Service, namun berbeda dengan Platform Thread, pada Virtual Thread, Executor Service yang disediakan hanya satu, hal ini karena memang kita tidak perlu lagi mengelola Thread nya secara manual
- Kita cukup memberikan tugas Runnable nya saja ke Virtual Thread Executor Service, dan semua akan dijalankan menggunakan Virtual Thread
- Kita bisa menggunakan `Executors.newVirtualThreadPerTaskExecutor()`



## Kode : Virtual Thread Executor Service

```
public class VirtualThreadExecutorApp {  
  
    public static void main(String[] args) throws IOException {  
        ExecutorService executor = Executors.newVirtualThreadPerTaskExecutor();  
        for (int i = 0; i < 10000; i++) {  
            executor.execute(() -> {  
                try {  
                    Thread.sleep(Duration.ofSeconds(2));  
                    System.out.println("Hello " + Thread.currentThread() + "!");  
                } catch (InterruptedException e) {  
                    throw new RuntimeException(e);  
                }  
            });  
        }  
  
        System.in.read();  
    }  
}
```