

Ringkasan Apa itu Multi-Threading?

Multi-threading merupakan teknik menjalankan beberapa tugas secara bersamaan dalam satu program untuk meningkatkan efisiensi dan mengurangi waktu eksekusi. Teknik ini digunakan untuk meningkatkan performa aplikasi yang melakukan banyak tugas sekaligus seperti menangani permintaan jaringan, pemrosesan data dan interaksi antar bagian program.

```
import java.text.SimpleDateFormat;

import java.util.Date;

public class Stopwatch {

    private static boolean running = false;

    private static long startTime;

    private static long stopTime;

    public static void start() {

        if (!running) {

            running = true;

            startTime = System.currentTimeMillis();

            new Thread(() -> {

                while (running) {

                    System.out.println(getElapsedTime());

                    try {

                        Thread.sleep(1000);
```

```
} catch (InterruptedException e) {

    e.printStackTrace();

}

}

}).start();

}

}

public static void stop() {

    if (running) {

        running = false;

        stopTime = System.currentTimeMillis();

    }

}

public static String getElapsedTime() {

    long elapsed;

    if (running) {

        elapsed = System.currentTimeMillis() - startTime;

    } else {

        elapsed = stopTime - startTime;

    }

    return new SimpleDateFormat("hh:mm:ss").format(new Date(elapsed));
}
```

```
}
```

```
public static void main(String[] args) {
```

```
    start();
```

```
}
```

```
}
```

```

import java.text.SimpleDateFormat;
import java.util.Date;

public class Stopwatch {
    private static boolean running = false;
    private static long startTime;
    private static long stopTime;

    public static void start() {
        if (!running) {
            running = true;
            startTime = System.currentTimeMillis();
            new Thread(() -> {
                while (running) {
                    System.out.println(getElapsedTime());
                    try {
                        Thread.sleep(1000);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            }).start();
        }
    }

    public static void stop() {
        if (running) {
            running = false;
            stopTime = System.currentTimeMillis();
        }
    }

    public static String getElapsedTime() {
        long elapsed;
        if (running) {
            elapsed = System.currentTimeMillis() - startTime;
        } else {
            elapsed = stopTime - startTime;
        }
        return new SimpleDateFormat("hh:mm:ss").format(new Date(elapsed));
    }

    public static void main(String[] args) {
        start();
    }
}

```

Lebih kurangnya, mohon maaf jika penjelasan saya kurang dimengerti, terima kasih.

n.b : Akan saya lampirkan Source-Code `Stopwatch.java`.

