

Analisa Penggunaan try, catch, dan throw

1. Blok throw:

- Pada metode **demo()**, sebuah objek **NullPointerException** diciptakan dan diberi pesan "Coba Throw".
- **throw t**; digunakan untuk memunculkan exception **NullPointerException**.
- Baris kode setelah **throw t**; (yaitu **System.out.println("Ini tidak lagi dicetak");**) tidak akan pernah dieksekusi karena exception telah dilempar dan eksekusi keluar dari metode **demo()**.

2. Blok try dalam main:

- Blok **try** digunakan untuk memanggil metode **demo()**.
- Jika **demo()** melempar exception, eksekusi segera beralih ke blok **catch** yang sesuai, dan baris **System.out.println("Selesai");** tidak akan dieksekusi.

3. Blok catch (NullPointerException e):

- Blok ini menangani **NullPointerException** yang dilempar dari metode **demo()**.
- Di dalam blok **catch**, pesan error dari exception dicetak menggunakan **System.out.println("Ada pesan error: " + e);**.
- **e** mencetak informasi tentang exception, termasuk pesan yang diberikan saat exception dibuat ("Coba Throw").

Kesimpulan

- Program ini menunjukkan bagaimana **throw** digunakan untuk memunculkan exception secara eksplisit.
- Blok **try** digunakan untuk melindungi kode yang mungkin melempar exception.
- Blok **catch** menangkap dan menangani exception tersebut, mencegah program crash dan memberikan informasi tentang error yang terjadi.
- Baris kode setelah **throw** tidak dieksekusi, yang menunjukkan bagaimana flow control dialihkan saat exception dilempar.