

Nama : Annisa Tasya A.

NIM : 20220040274

Kelas : TI22A

---

## Tugas 10

### Percobaan 5

Analisa penggunaan try dan catch

Penggunaan **try** and **catch** pada perogram ini dapat melanjutkan eksekusi meskipun ada pengecualian terjadi, sehingga memeberikan kontrol yang lebih baik atas alur program dan menghindari berhentinya program secara tiba-tiba.

### Percobaan 6

Analisa penggunaan try dan catch

Untuk menangani pengecualian yang dilempar oleh method '**demo()**'

- **Demo()**

Untuk menganalisis objek '**NullPointerException**' dengan menggunakan pesan '**Coba Throw**', lalu '**demo()**' akan melemparkan pengecualian tersebut menggunakan pernyataan '**throw**'.

- **Try-Catch didalam main()**

Pemanggilan '**demo()**' dilakukan dalam blok '**try**' jika pengecualian '**NullPointerException**' dilempar oleh '**demo()**' maka '**catch**' akan menangkapnya agar pesan error tidak muncul.

### Percobaan 7

Analisa penggunaan try and catch

Penggunaan blok ini untuk menangani sebuah pengecualian.

- **Try**

Ada sebuah pernyataan berupa '**throw new Exception ("Here's my Exception")**' dimana secara eksplisit melemparkan objek '**Exception**', untuk membuat sebuah

pengecualian. Dan saat **'throw new Exception ("Here's my Exception")'** dijalankan program ini akan mengeluarkan blok **'try'** dan tidak mengeksekusi pernyataan berikutnya.

- **Catch**

Pada **'catch (Exception e)'** akan menangkap objek pengecualian yang akan dilempar ke dalam blok **'try'** lalu pengecualian akan ditangkap blok **'catch'** untuk dieksekusi dan mencetak pesan **'Caught Exception'**. Dan kita mencetak informasi tentang pengecualian yang ditangkap menggunakan beberapa metode. Seperti **'e.getMessage()'** mencetak pesan yang ada pada pengecualian, dan **'e.toString()'** mencetak representasi String pengecualian yang ada nama kelas dan pesan, sedangkan **'e.printStackTrace()'** mencetak stack trace pengecualian begitu juga dengan detail lokasi pengecualian terjadi.

## **Percobaan 8**

Analisa penggunaan throws

**Throw** pada java digunakan untuk memberi tahu pemanggil bahwa sebuah metode bisa melempar pengecualian tertentu. Jadi pada codingan ini **'throws IOException'** pada sebuah deklarasi **'methodB'** itu untuk memberi tahu pemanggil bahwa method tersebut melempar pengecualian tipe **'IOException'**. Jadi pemanggilan **'methodB'** harus menangani pengecualian tersebut seperti pengecualian dalam blok **'try-Catch'** pada metode **'main'**.