**TUGAS SESI 10**

**EXCEPTION HANDLING**

Nama : Syefin Fazri Nugraha

NIM : 20210040097

Kelas : TI23F

Mata Kuliah : Pemrograman Berorientasi Objek

Dosen Pengampu : Alun Sujjada,S.Kom., M.T

|  |  |  |  |
| --- | --- | --- | --- |
| **Analisa Percobaan** | **Masalah** | **Solusi** | **Hasil** |
| **Percobaan 1** | Akses indeks array di luar batas (a[5] pada array dengan indeks 0–4). | Menangkap Exception dengan blok try – catch untuk mencegah program berhenti mendadak. | Program tetap berjalan dan menampilkan pesan error. |
| **Percobaan 2** | Perulangan while melebihi batas array greetings. | Menangkap ArrayIndexOutOfBoundsException dan mengatur ulang indeks. | Perulangan tetap berjalan tanpa error meski terjadi indeks berlebih. |
| **Percobaan 3** | Pembagian bilangan dengan nol (bil/0). | Menangkap ArithmeticException untuk mencegah program terhenti. | Pesan error ditangani dan program tetap berjalan. |
| **Percobaan 4** | Kesalahan urutan instruksi di blok try, menyebabkan exception pertama menghentikan eksekusi. | Menangani error berdasarkan urutan yang benar, memisahkan penanganan ArithmeticException dan ArrayIndexOutOfBoundsException. | Error dapat diidentifikasi dan ditangani sesuai jenisnya. |

**Percobaan 5**

* **Fokus:** Menampilkan informasi error secara detail dengan:
  + e.getMessage()
  + e.printStackTrace()
* **Hasil:** Program menampilkan pesan error, jejak error, dan tetap berjalan.

**Percobaan 6**

* **Fokus:** Penggunaan throw secara manual di method demo().
* **Hasil:** Method demo melempar NullPointerException yang ditangkap di main. Baris
* setelah throw tidak dieksekusi.

**Percobaan 7**

* **Fokus:** Melempar exception Exception secara manual.
* **Hasil:** Exception berhasil ditangkap, dan program menampilkan pesan detail dan stack trace.

**Percobaan 8**

* **Fokus:** Penggunaan throws pada method yang mungkin menghasilkan exception (methodB).
* **Hasil:** Program dapat menangani error dengan try – catch atau mendeklarasikan throws di main. Versi kedua menggunakan finally untuk memastikan blok tertentu tetap dijalankan.

**Percobaan 9**

* **Fokus:** Melempar exception jika string kosong saat dibalik (reverse).
* **Hasil:** Jika string kosong, method reverse melempar exception dan ditangkap di main. Blok finally tetap dijalankan.

**Percobaan 10**

* **Fokus:** Penanganan IOException saat menulis dan membaca file dengan RandomAccessFile.
* **Hasil:** Program menangani file I/O dan jika terjadi kesalahan indeks, error dapat diantisipasi.

**Percobaan 11**

* **Fokus:** Membuat exception kustom RangeErrorException yang mewarisi Throwable.
* **Hasil:** Exception kustom berhasil dilempar dan ditangkap, menampilkan pesan yang sesuai.

**Percobaan 12**

* **Fokus:** Membuat exception kustom MyException yang mewarisi Exception.
* **Hasil:** Exception kustom di-throw jika kondisi tertentu terpenuhi. Ditangkap di main dan ditampilkan pesan tangkapannya.

**Kesimpulan**

Dari percobaan di atas, dapat disimpulkan:

* Exception handling penting untuk mencegah program berhenti mendadak akibat error.
* try-catch digunakan untuk menangkap error.
* throw digunakan untuk melempar error secara manual.
* throws digunakan untuk mendeklarasikan bahwa method dapat menghasilkan exception.
* Pembuatan exception kustom memungkinkan developer menangani error spesifik sesuai kebutuhan aplikasi.