TUGAS 10

PRAKTIKUM EXCEPTION

Nama : **Rendi Alfa Nayoan**

Nim : **20230040210**

Kelas : **TI23F**

Mata Kuliah : **Pemrograman Berorientasi Objek**

PERCOBAAN 1

Kode: Akses indeks ke-5 pada array yang hanya berisi 5 elemen (indeks 0–4)

Try: Menyimpan perintah a[5] = 100;

Catch: Menangani ArrayIndexOutOfBoundsException

Analisa: Program mencoba mengakses memori di luar batas array. try-catch mencegah program crash dan menampilkan pesan error.

PERCOBAAN 2

Kode: Melakukan perulangan hingga i = 3, padahal array hanya punya 3 elemen (indeks 0–2)

Try: Mencetak isi array

Catch: Menangani ArrayIndexOutOfBoundsException, dan mereset i = 0

Analisa: Saat indeks melebihi batas, program tidak berhenti, tapi mengulang dari awal. Simulasi error dan pemulihan otomatis.

PERCOBAAN 3

Kode: System.out.println(bil / 0);

Try: Melakukan pembagian dengan nol

Catch: Menangani ArithmeticException

Analisa: Error aritmatika (pembagian dengan nol) dicegah agar program tetap berjalan. Error ditangani secara umum (Exception).

PERCOBAAN 4:

Kode: Menjalankan dua error: b[3] (indeks salah) dan bil/0

Try: Menjalankan dua baris error

Catch: Menangani ArithmeticException, ArrayIndexOutOfBoundsException, dan fallback Exception

Analisa: Urutan kode menentukan error mana yang muncul lebih dulu. Masing-masing error punya catch khusus.

PERCOBAAN 5:

Kode: System.out.println(bil / 0);

Try: Menyebabkan ArithmeticException

Catch: Menangkap error dan mencetak getMessage() serta printStackTrace()

Analisa: Program tidak hanya menangkap error tapi juga menampilkan informasi mendetail tentang sumber error.

PERCOBAAN 6:

Kode: Melempar NullPointerException secara manual

Try: Memanggil method demo() yang melempar error

Catch: Menangani NullPointerException

Analisa: Contoh penggunaan throw untuk membuat error sendiri. Baris setelah throw tidak dijalankan.

PERCOBAAN 7:

Kode: throw new Exception(...)

Try: Langsung melempar objek Exception

Catch: Menangkap error dan mencetak berbagai metode (getMessage(), toString(), printStackTrace())

Analisa: Memberikan contoh struktur informasi yang bisa diambil dari sebuah Exception.

PERCOBAAN 8:

Kode: Method methodB() melempar error aritmatika tapi dideklarasikan throws IOException

Try: Memanggil methodB() dan menangkap error

Catch: Menangani Exception secara umum

Finally: Selalu dijalankan untuk mencetak pesan

Analisa: Error tetap ditangani meskipun exception yang terjadi berbeda dari yang dideklarasikan. finally memastikan kode selalu dijalankan.

PERCOBAAN 9:

Kode: Membalik string, jika string kosong maka lempar Exception

Try: Memanggil method reverse(...)

Catch: Menangani Exception jika string kosong

Finally: Mencetak pesan akhir

Analisa: Mencegah error jika string tidak valid, dan menjamin finally dijalankan apapun hasilnya.

PERCOBAAN 10:

Kode: Menggunakan RandomAccessFile untuk menulis dan membaca file

Try: Membuka file dan menulis data

Catch: Menangani IOException

Analisa: Melindungi proses file agar tidak error jika terjadi kesalahan file akses atau indeks file yang diluar batas.

PERCOBAAN 11:

Kode: Melempar error custom RangeErrorException yang extends Throwable

Try: Mengecek position > 0 lalu melempar exception

Catch: Menangani exception custom dan mencetak pesan

Analisa: Menunjukkan cara membuat dan menangani exception buatan sendiri menggunakan Throwable.

PERCOBAAN 12:

Kode: Melempar exception custom MyException jika nama "amir"

Try: Memanggil method tampil()

Catch: Menangani MyException

Analisa: Menggabungkan pengecekan kondisi, pembuatan exception sendiri, dan penanganannya.