KUSUMA WEB ACADEMY

DASAR-DASAR JALUR JAVA

Modul 14: Keandalan Sistem - Penanganan Error (Exception Handling)

Hak Cipta © 2026 Kusuma Web. All Rights Reserved.

Memahami Kejadian Tak Terduga (Exception)

Apa itu Exception?

- Gangguan tidak normal atau galat (*runtime error*) yang memicu matinya program di tengah eksekusi akibat kondisi eksternal tak valid.
- **Contoh klasifkasi:**
  - Pembagian angka dengan nol (`ArithmeticException`).
  - Membaca file yang tidak ada (`FileNotFoundException`).

Kategori Exception

1. **Checked Exception:** Galat yang dideteksi sejak kompilasi. Wajib dideklarasikan atau ditangkap di program.
2. **Unchecked (Runtime) Exception:** Terjadi saat eksekusi berjalan akibat cacat logika kode penulisan.

Kusuma Web

Mekanisme Penangkapan Galat: `try-catch-finally`

Blok Penyusun Struktur

- **Try:** Menampung baris kode berbahaya yang berpotensi memicu timbulnya exception.
- **Catch:** Kode eksekusi cadangan penanganan jika exception benar-benar terjadi.
- **Finally:** Blok yang **pasti berjalan** baik ada exception maupun tidak (bagus untuk aksi penutupan database/file).

```
try {
    int hasil = 10 / 0; // Arithmetic
        Error!
} catch (ArithmeticException e) {
    System.out.println("Eror: Gagal
        membagi!");
} finally {
    System.out.println("Pembersihan
        memori selesai.");
}
```

Melacak & Meneruskan Galat: Keyword `throw` vs `throws`

Keyword `throw`

- Digunakan untuk melemparkan (*trigger*) exception secara sengaja dari dalam tubuh logika program.

```
void cekUmur(int umur) {
    if (umur < 18) {
        throw new ArithmeticException("
            Belum dewasa!");
    }
}
```

Keyword `throws`

- Diletakkan pada deklarasi *signature* method untuk memperingatkan pemanggilnya bahwa method ini berpotensi melempar exception tertentu.

```
void bacaBerkas() throws IOException {
    // Berkas berpotensi tidak ada
}
```

Membuat Exception Mandiri (Custom Exception)

Kelebihan Custom Exception

- Membantu pemetaan error bisnis yang spesifik dan mudah dipahami, tidak sekadar error teknis standar bawaan bahasa pemrograman Java.
- Untuk membuatnya, cukup buat class baru yang mengekstensi `Exception` (checked) atau `RuntimeException` (unchecked).

```
// Membuat Exception Kustom
class SaldoKurangException extends
    Exception {
    public SaldoKurangException(String
        pesan) {
        super(pesan);
    }
}

// Cara Penggunaan
public class Bank {
    void ambilTunai(int saldo, int
        tagihan)
         throws SaldoKurangException {
        if (tagihan > saldo) {
            throw new
                SaldoKurangException("
                Tabungan kurang!");
        }
    }
}
```

Kusuma