LAPORAN TUGAS TEKNIK PEMRORGAMAN

PERTEMUAN 6

****

NAMA: FAUZAN JUNIAR MULYANA

NIM: 241524011

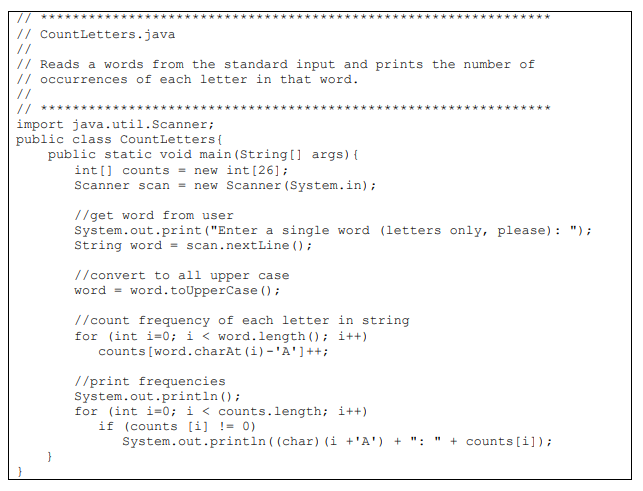
**POLITEKNIK NEGERI BANDUNG**

**PROGRAM STUDI D4-TEKNIK INFORMATIKA**

**2024**

# **Case 1: Exceptions Aren't Always Errors**

Kode sebelumnya:



Kode setelah dimodified:



Analisis Kesalaham:

**Masalah**: Kode asli mengonversi karakter input ke indeks array tanpa validasi, mengabaikan risiko karakter non-huruf seperti spasi atau simbol. Ini menyebabkan ArrayIndexOutOfBoundsException karena indeks dihitung dengan mengurangkan 'A', yang menghasilkan nilai negatif atau di luar batas 0-25 untuk karakter invalid.

**Penyebab Utama**: Kurangnya pengecekan prasyarat (*precondition*) pada input sebelum pemrosesan.

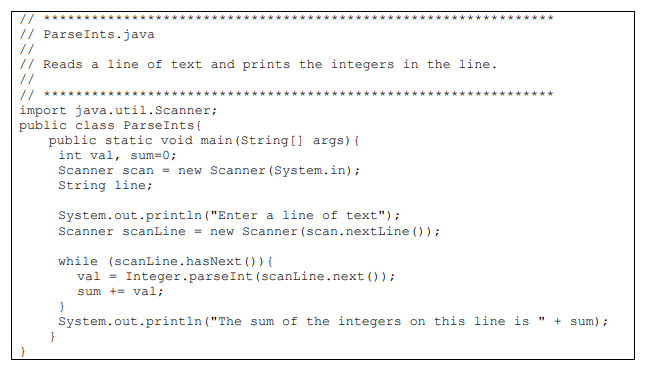
**Solusi**: Memasukkan operasi akses array ke dalam blok try untuk menangkap exception, sengaja mengabaikan karakter non-huruf tanpa menghentikan program.

**Perubahan Kritis**: Penerapan *defensive coding* dengan memisahkan logika validasi dari logika bisnis menggunakan try-catch.

**Hasil**: Program tetap stabil meskipun input mengandung karakter tidak valid, dengan pesan error yang informatif untuk debugging.

# **Case 2: Placing Exception Handlers**

Kode Sebelum:



Kode setelah dimodified:



Analisis Kesalaham:

**Masalah**: Penempatan blok try di luar loop menyebabkan penjumlahan berhenti pada token non-integer pertama, karena exception yang terjadi menghentikan eksekusi loop sepenuhnya.

**Penyebab Utama**: Struktur penanganan error yang terlalu luas, tidak memisahkan risiko per iterasi.

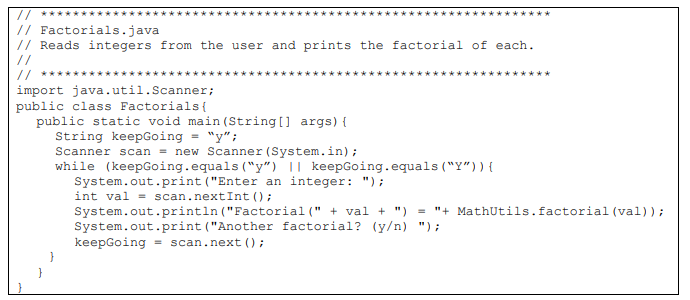
**Solusi**: Memindahkan try-catch ke dalam loop untuk mengisolasi exception per token, memastikan iterasi berikutnya tetap berjalan meskipun ada token invalid.

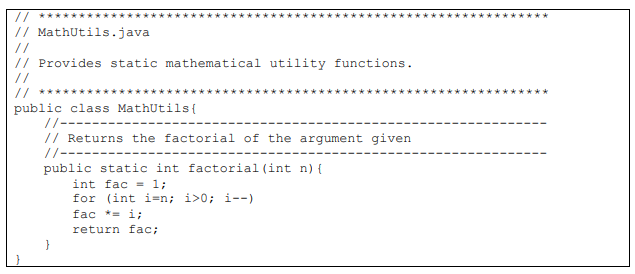
**Perubahan Kritis**: Menghindari *error propagation* dengan membatasi scope exception handler ke level operasi terkecil (per token).

**Hasil**: Program memproses seluruh input, menjumlahkan semua integer yang valid dan mengabaikan token non-integer secara elegan.

# **Case 3: Throwing Exceptions**

Kode Sebelum:





Kode setelah dimodified:





Analisis Kesalahan:

**Masalah**: Metode factorial tidak membatasi input negatif atau nilai besar (>16), menghasilkan output matematis yang tidak valid (1 untuk bilangan negatif, overflow untuk nilai besar).

**Penyebab Utama**: Tidak adanya validasi prasyarat (*precondition*) pada argumen metode.

**Solusi**: Melempar IllegalArgumentException dengan pesan deskriptif saat input tidak memenuhi kriteria valid (0 ≤ n ≤ 16).

**Perubahan Kritis**: Mengubah pendekatan "diam-diam mengembalikan nilai salah" menjadi *fail-fast* dengan exception untuk memaksa penanganan error.

**Hasil**: Pengguna mendapat feedback jelas tentang kesalahan input, dan program menghindari perhitungan tidak bermakna, meningkatkan keandalan logika bisnis.