**LAPORAN PRAKTIKUM TEKNIK PEMROGRAMAN :**

**DEFENSIVE PROGRAMMING**

A person wearing a black head scarf

Description automatically generatedA blue and orange hexagon with a white circle and a white circle

Description automatically generated

**POLITEKNIK NEGERI BANDUNG**

**FAKULTAS TEKNIK KOMPUTER DAN INFORMATIKA**

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**

Jl. Gegerkalong Hilir, Ciwaruga, Kec. Parongpong, Kabupaten Bandung Barat, Jawa Barat 40559

# **DAFTAR ISI**

[DAFTAR ISI i](#_Toc193363685)

[A. Case 1 1](#_Toc193363686)

[1. Tujuan 1](#_Toc193363687)

[2. Implementasi 1](#_Toc193363688)

[2.1 Kode sebelum diperbaiki 1](#_Toc193363689)

[2.2 Kode setelah diperbaiki 2](#_Toc193363690)

[3. Output 3](#_Toc193363691)

[4. Kesimpulan 3](#_Toc193363692)

[B. Case 2 3](#_Toc193363693)

[1. Tujuan 3](#_Toc193363694)

[2. Implementasi 4](#_Toc193363695)

[2.3 Kode sebelum diperbaiki 4](#_Toc193363696)

[2.4 Kode setelah diperbaiki 4](#_Toc193363697)

[3. Output 5](#_Toc193363698)

[4. Kesimpulan 6](#_Toc193363699)

[C. Case 3 6](#_Toc193363700)

[1. Tujuan 6](#_Toc193363701)

[2. Implementasi 6](#_Toc193363702)

[2.5 Kode sebelum diperbaiki 6](#_Toc193363703)

[2.6 Kode setelah diperbaiki 7](#_Toc193363704)

[3. Output 9](#_Toc193363705)

[4. Kesimpulan 9](#_Toc193363706)

[Link GitHub 9](#_Toc193363707)

# **Case 1**

## Tujuan

Program ini bertujuan untuk menangani pengecualian (exception) dalam Java menggunakan try-catch. Program CountLetters.java akan membaca sebuah kata dari input user, mengonversinya menjadi huruf besar, lalu dapat menghitung jumlah kemunculan setiap huruf dalam kata tersebut. Namun, jika pengguna memasukkan lebih dari satu kata atau karakter non-huruf (seperti spasi, angka, atau simbol), program akan mengalami **ArrayIndexOutOfBoundsException**.

## Implementasi

### Kode sebelum diperbaiki

Kode awal tidak memiliki mekanisme error handling, sehingga jika pengguna memasukkan karakter selain huruf, program akan mengalami error (ArrayIndexOutOfBoundsException).

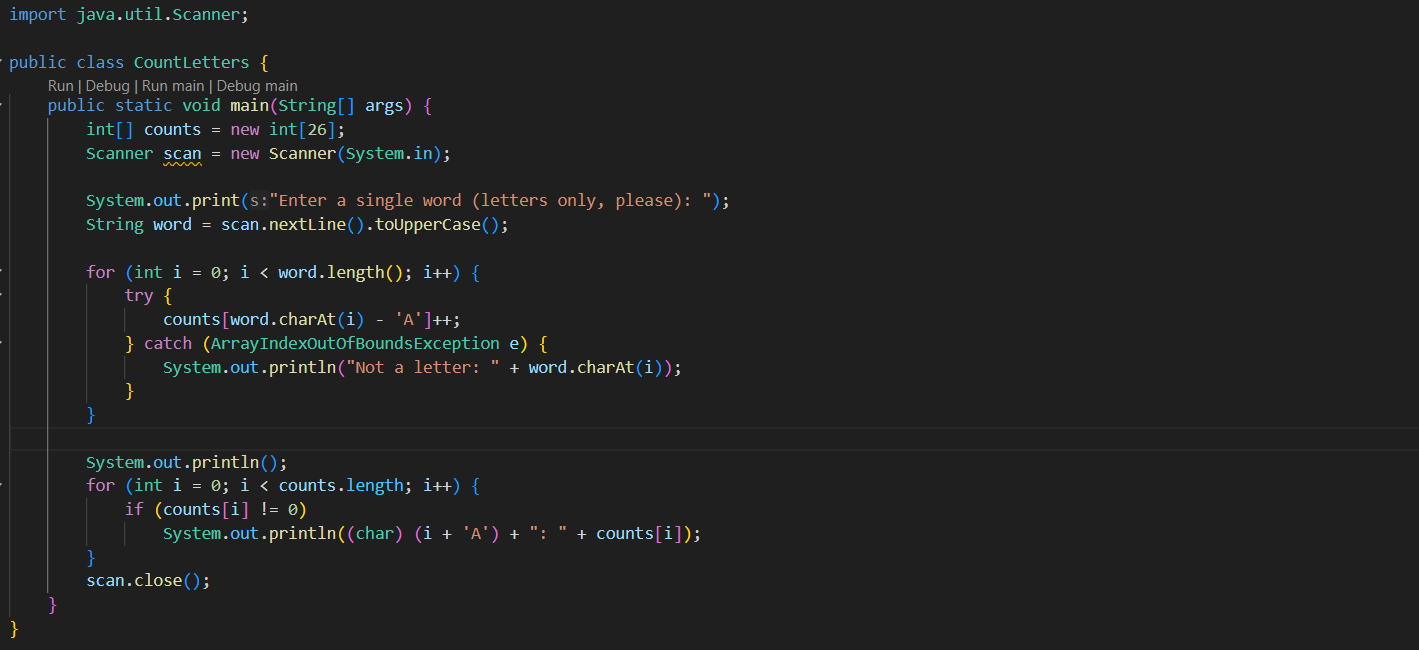
A screen shot of a computer code

AI-generated content may be incorrect.Pada program ini terdapat beberapa maslaah, diantaranya :

1. Jika pengguna memasukkan karakter selain huruf (misalnya Hello! atau 123), program akan crash dengan ArrayIndexOutOfBoundsException.
2. Tidak ada penanganan error untuk karakter non-huruf.

### Kode setelah diperbaiki

Kode diperbaiki dengan menambahkan try-catch di dalam loop untuk menangani exception dan menampilkan pesan yang lebih informatif jika ditemukan karakter non-huruf.



Perbaikan yang telah saya lakukan, diantaranya :

1. Try-catch di dalam loop untuk menangani ArrayIndexOutOfBoundsException.
2. Jika ditemukan karakter selain huruf, program tidak crash, tetapi menampilkan pesan "Not a letter: X".
3. Program tetap melanjutkan prosesnya untuk karakter yang valid.

## Output

A computer screen with white text

AI-generated content may be incorrect.

## Kesimpulan

Sebelum perbaikan, program tidak dapat menangani karakter selain huruf, sehingga terjadi error jika input mengandung angka atau simbol. Dengan menerapkan defensive programming, program menjadi lebih tangguh dan tetap berjalan meskipun pengguna memasukkan input yang tidak sesuai. Penggunaan **try-catch** memungkinkan penanganan error yang lebih baik, sekaligus memberikan informasi yang lebih jelas kepada pengguna.

# **Case 2**

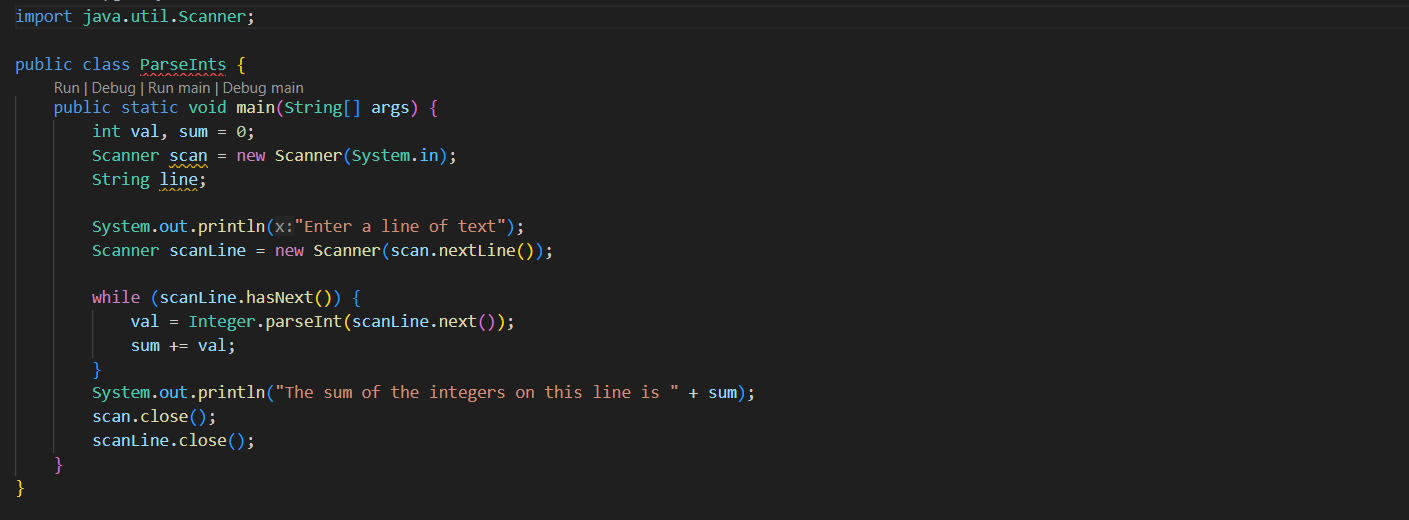
## Tujuan

Kasus ini bertujuan untuk menangani kesalahan NumberFormatException yang terjadi ketika program mencoba mengonversi teks non-angka menjadi integer. Program ParseInts.java membaca satu baris input dari pengguna, mengekstrak angka di dalamnya, dan menjumlahkan angka tersebut.

## Implementasi

### Kode sebelum diperbaiki

Kode awal tidak memiliki mekanisme error handling, sehingga jika pengguna memasukkan teks yang bukan angka, program akan mengalami error (NumberFormatException).



Dari program diaatas terdapat beberapa masalah, dianataranya:

1. Jika input mengandung kata atau karakter non-angka (misalnya "We have 2 dogs and 1 cat"), program crash karena NumberFormatException.
2. Program tidak memiliki mekanisme untuk melewati input yang salah dan tetap menjumlahkan angka yang valid.

### Kode setelah diperbaiki

Kode diperbaiki dengan menambahkan try-catch di dalam loop agar program dapat mengabaikan input yang tidak valid tanpa crash.

A computer screen shot of text

AI-generated content may be incorrect.

Perbaikan yang telah saya lakukan, diantaranya:

1. Try-catch di dalam loop untuk menangani NumberFormatException.
2. Jika ditemukan input yang bukan angka, program melewati input tersebut tanpa crash.
3. Pesan "Skipping non-integer value" akan muncul saat program menemukan input non-angka.

## Output

A black screen with white text

AI-generated content may be incorrect.

## Kesimpulan

Sebelum perbaikan, program tidak dapat menangani input non-angka dan langsung crash ketika menemukan teks dalam masukan pengguna. Dengan menerapkan defensive programming, program dapat melewati input yang tidak valid dan tetap melanjutkan perhitungan angka yang benar. Penggunaan try-catch di dalam loop memungkinkan setiap kata diproses secara individual, sehingga jika terjadi kesalahan, program tetap dapat melanjutkan ke kata berikutnya tanpa terhenti. Perbaikan ini menjadikan program tidak mudah crash meskipun pengguna memasukkan data dengan format yang tidak sesuai.

# **Case 3**

## Tujuan

Program ini diperbaiki untuk mencegah perhitungan faktorial pada angka negatif, karena secara matematis faktorial tidak didefinisikan untuk bilangan negatif. Selain itu, perbaikan juga dilakukan untuk mencegah overflow saat menghitung faktorial dari angka yang terlalu besar (>16), karena tipe data int tidak mampu menyimpan hasilnya. Dengan penanganan error yang lebih baik, program tetap stabil dan tidak mengalami crash, serta memberikan pesan yang jelas kepada pengguna jika input yang dimasukkan tidak valid.

## Implementasi

### Kode sebelum diperbaiki

Kode awal tidak memiliki mekanisme error handling, sehingga jika pengguna memasukkan angka negatif atau lebih dari 16, program akan memberikan hasil yang salah atau mengalami overflow.

A computer screen shot of a computer code

AI-generated content may be incorrect.

A black screen with colorful text

AI-generated content may be incorrect.

Dari program diatas terdapat beberapa masalah, diantaranya :

1. Jika pengguna memasukkan angka negatif, program mengembalikan 1, padahal seharusnya faktorial tidak didefinisikan untuk angka negatif.
2. Jika pengguna memasukkan angka lebih dari 16, terjadi overflow, sehingga hasilnya salah (angka negatif karena batas int).
3. Tidak ada mekanisme error handling untuk input yang tidak valid.

### Kode setelah diperbaiki

Kode diperbaiki dengan menambahkan try-catch di dalam Factorials.java dan melempar IllegalArgumentException di MathUtils.factorial() untuk menangani input yang tidak valid.

A screenshot of a computer

AI-generated content may be incorrect.

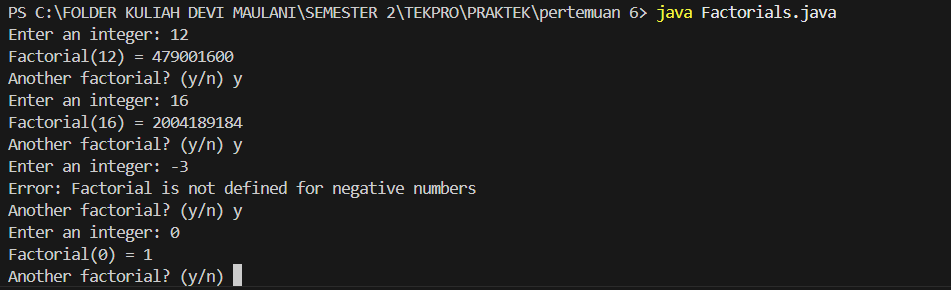
A computer code on a black background

AI-generated content may be incorrect.

Perbaikan yang telah saya lakukan, diantaranya:

1. Pada Try-catch di Factorials.java untuk menangani error dan menampilkan pesan yang jelas.
2. Throw IllegalArgumentException di MathUtils.factorial() jika input negatif atau lebih dari 16.
3. Program tidak crash dan tetap berjalan dengan baik meskipun input tidak valid.

## Output



## Kesimpulan

Sebelum perbaikan, program tidak menangani input negatif dan angka yang terlalu besar dengan benar, sehingga menghasilkan output yang salah atau mengalami overflow. Dengan menerapkan defensive programming, program dapat mendeteksi dan menangani input yang tidak valid menggunakan IllegalArgumentException. Penggunaan try-catch di dalam Factorials.java memastikan bahwa program tetap berjalan meskipun terjadi kesalahan input, tanpa menyebabkan crash. Perbaikan ini menjadikan program mampu memberikan pesan error yang jelas kepada pengguna.

# **Link GitHub**