

LAPORAN PRAKTIKUM

Exception

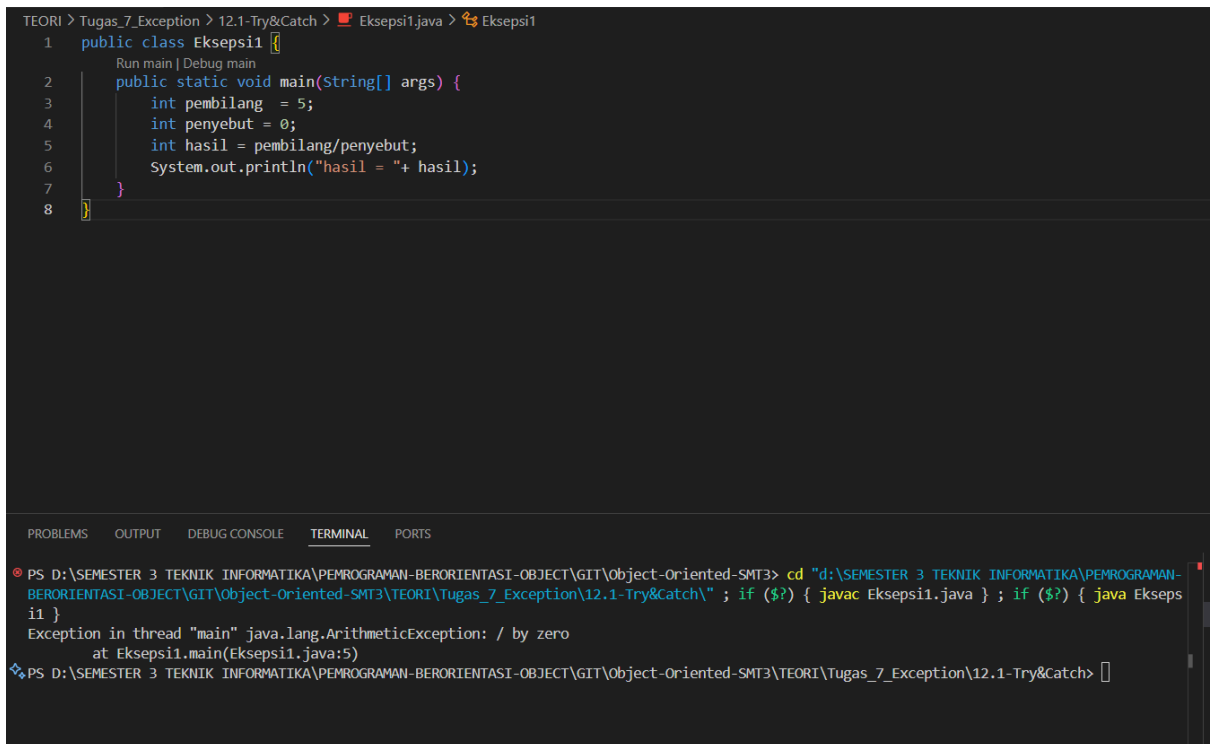


Disusun Oleh:
Dzakir Tsabit Asy Syafiq (241511071)
Jurusan Teknik Komputer dan Informatika

Program Studi D-3 Teknik Informatika
Politeknik Negeri Bandung
02/10/2025

12.1 try – catch

Eksepsi1.java :



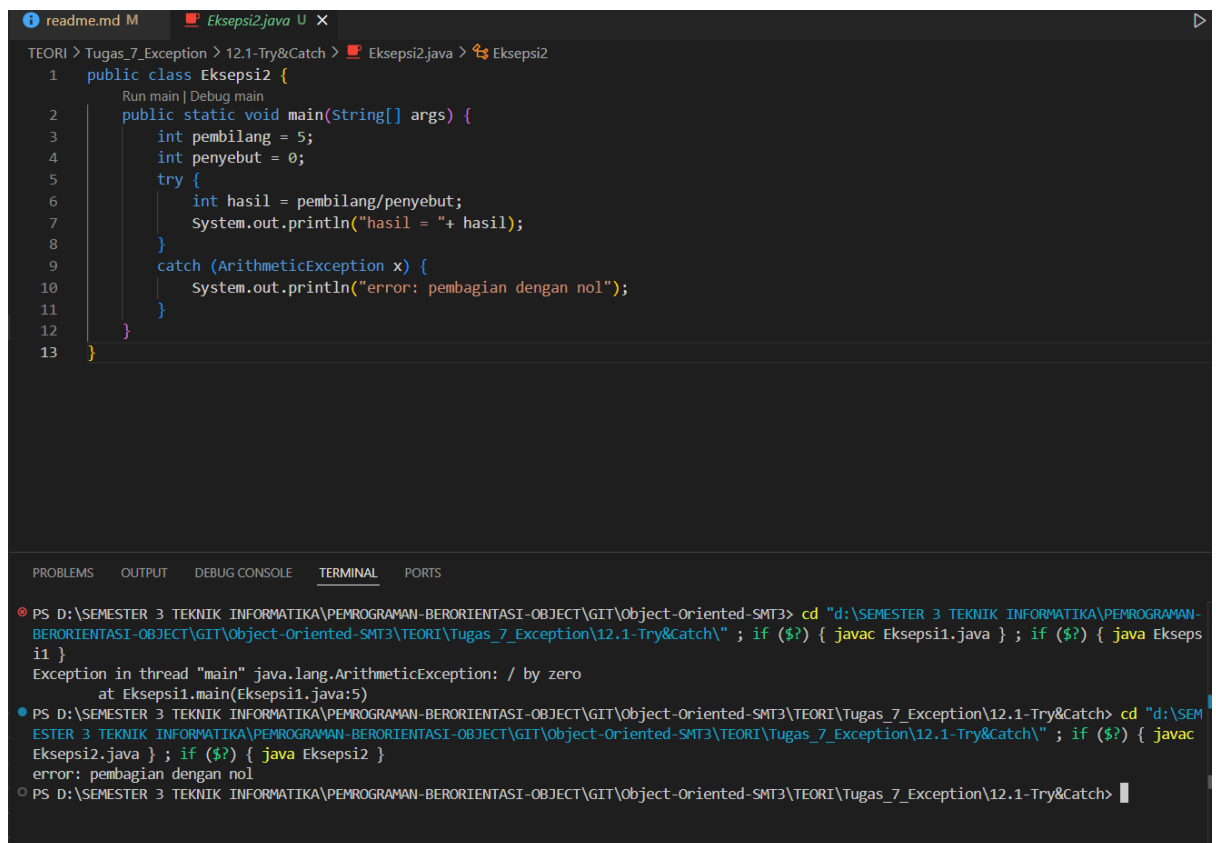
```
TEORI > Tugas_7_Exception > 12.1-Try&Catch > Eksepsi1.java > Eksepsi1
1 public class Eksepsi1 {
2     public static void main(String[] args) {
3         int pembilang = 5;
4         int penyebut = 0;
5         int hasil = pembilang/penyebut;
6         System.out.println("hasil = " + hasil);
7     }
8 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI-OBJECT\GIT\Object-Oriented-SMT3> cd "d:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI-OBJECT\GIT\Object-Oriented-SMT3\TEORI\Tugas_7_Exception\12.1-Try&Catch\" ; if ($?) { javac Eksepsi1.java } ; if ($?) { java Eksepsi1 }
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at Eksepsi1.main(Eksepsi1.java:5)
PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI-OBJECT\GIT\Object-Oriented-SMT3\TEORI\Tugas_7_Exception\12.1-Try&Catch>
```

Eksepsi2.java :

Setelah menggunakan try – catch



```
TEORI > Tugas_7_Exception > 12.1-Try&Catch > Eksepsi2.java > Eksepsi2
1 public class Eksepsi2 {
2     Run main | Debug main
3     public static void main(String[] args) {
4         int pembilang = 5;
5         int penyebut = 0;
6         try {
7             int hasil = pembilang/penyebut;
8             System.out.println("hasil = "+ hasil);
9         }
10        catch (ArithmeticException x) {
11            System.out.println("error: pembagian dengan nol");
12        }
13    }
}
```

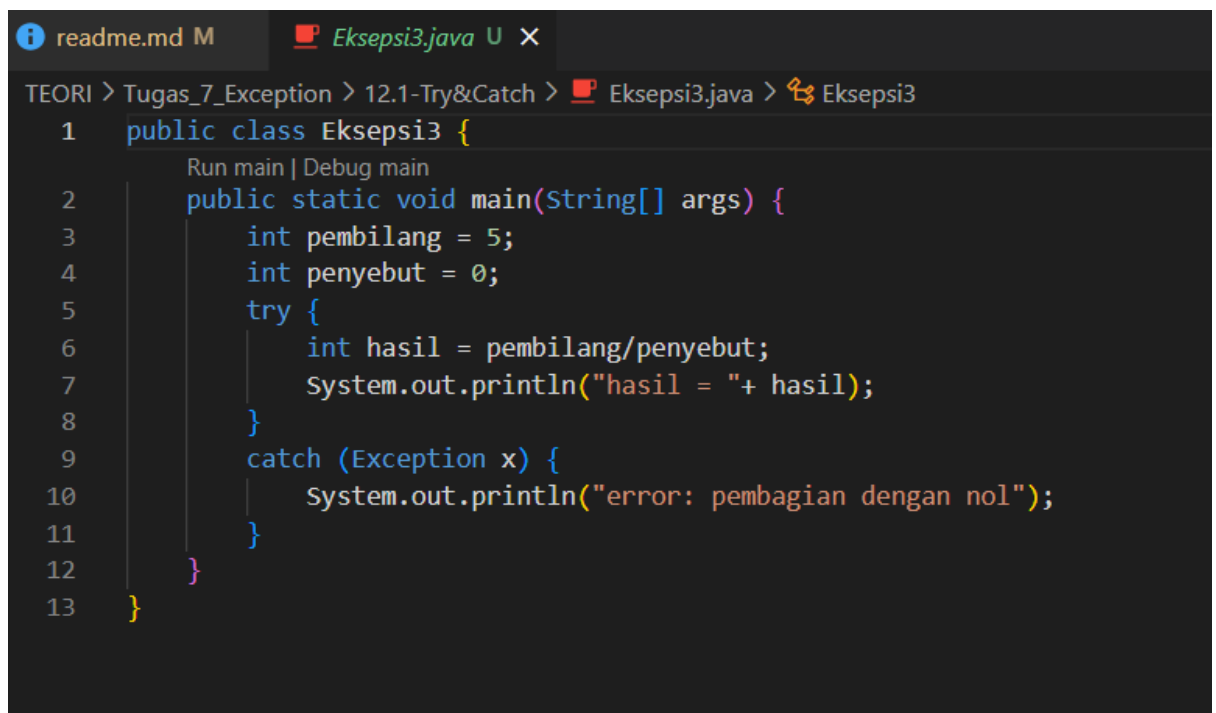
PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI-OBJECT\GIT\Object-Oriented-SMT3> cd "d:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI-OBJECT\GIT\Object-Oriented-SMT3\TEORI\Tugas_7_Exception\12.1-Try&Catch\" ; if ($?) { javac Eksepsi1.java } ; if ($?) { java Eksepsi1 }
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at Eksepsi1.main(Eksepsi1.java:5)
PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI-OBJECT\GIT\Object-Oriented-SMT3\TEORI\Tugas_7_Exception\12.1-Try&Catch> cd "d:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI-OBJECT\GIT\Object-Oriented-SMT3\TEORI\Tugas_7_Exception\12.1-Try&Catch\" ; if ($?) { javac Eksepsi2.java } ; if ($?) { java Eksepsi2 }
error: pembagian dengan nol
PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI-OBJECT\GIT\Object-Oriented-SMT3\TEORI\Tugas_7_Exception\12.1-Try&Catch>
```

Eksepsi3.java :

Menggunakan Try-Catch

Dapat juga menangkap error yang tidak hanya matematik saja seperti contoh di bawah ini jika tidak diketahui kemungkinan error nya :



```
TEORI > Tugas_7_Exception > 12.1-Try&Catch > Eksepsi3.java > Eksepsi3
1 public class Eksepsi3 {
2     Run main | Debug main
3     public static void main(String[] args) {
4         int pembilang = 5;
5         int penyebut = 0;
6         try {
7             int hasil = pembilang/penyebut;
8             System.out.println("hasil = "+ hasil);
9         }
10        catch (Exception x) {
11            System.out.println("error: pembagian dengan nol");
12        }
13    }
}
```

```

PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-
SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI
Eksepsi3.java } ; if ($?) { java Eksepsi3 }
error: pembagian dengan nol

```

Eksepsi4.java :

Tersedia method getMessage() yang berfungsi untuk menampilkan pesan sebenarnya yang terkandung pada eksepsi yang ditimbulkan. Method ini diletakkan pada objek eksepsi yang dilewatkan

```

TEORI > Tugas_7_Exception > 12.1-Try&Catch > Eksepsi4.java > Eksepsi4
1 public class Eksepsi4 {
    Run main | Debug main
2 public static void main(String[] args){
3     int pembilang = 5;
4     int penyebut = 0;
5     try {
6         int hasil = pembilang/penyebut;
7         System.out.println("hasil =" + hasil);
8     }
9     catch(Exception x) {
10        System.out.println(x.getMessage());
11    }
12    System.out.println("statemen setelah blok try catch");
13 }
14 }

```

```

error: pembagian dengan nol
PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-
SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI
Eksepsi4.java } ; if ($?) { java Eksepsi4 }
/ by zero
statemen setelah blok try catch
PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-

```

Eksepsi5.java :

Dengan cara diatas, dapat diketahui dengan pasti apa kesalahan yang terjadi. Jika ingin mengeluarkan informasi stack trace (stack trace adalah daftar dari semua method yang terpanggil pada saat program menunjuk ke baris kode yang menimbulkan eksepsi) dapat menggunakan method printStackTrace(). Berikut contoh penggunaannya:

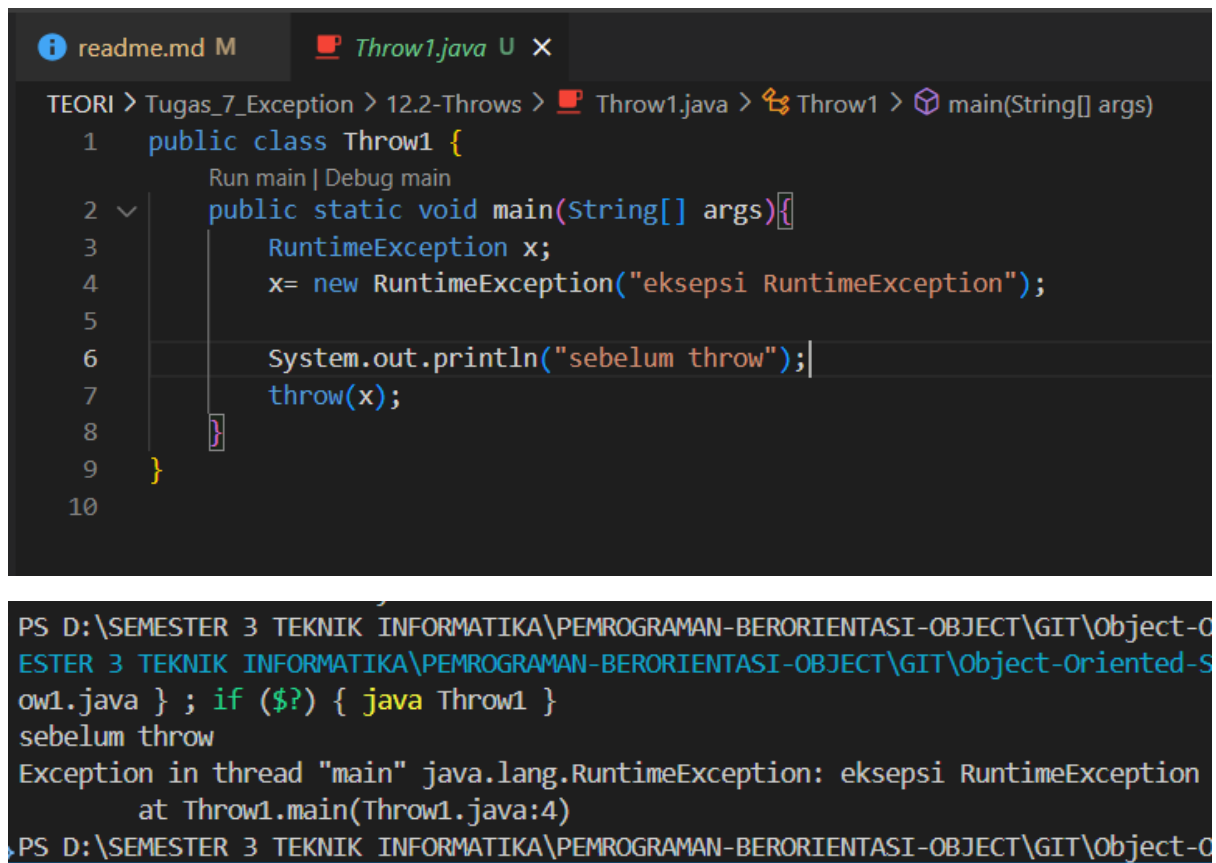
```
readme.md M Eksepsi5.java 1, U X
TEORI > Tugas_7_Exception > 12.1-Try&Catch > Eksepsi5.java > Eksepsi5
1 public class Eksepsi5 {
2     Run main | Debug main
3     public static void main(String[] args){
4         int pembilang = 5;
5         int penyebut= 0;
6         try {
7             int hasil = pembilang/penyebut;
8             System.out.println("hasil =" + hasil);
9         }
10        catch(Exception x) {
11            x.printStackTrace(); Print Stack Trace
12        }
13        System.out.println("statemen setelah blok try catch");
14    }
}
```

```
PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTA
Eksepsi5.java } ; if ($?) { java Eksepsi5 }
java.lang.ArithmeticException: / by zero
    at Eksepsi5.main(Eksepsi5.java:6)
statemen setelah blok try catch
PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-B
```

12.2 throw

Throw1.java :

Java menyediakan pernyataan yang digunakan untuk melontarkan eksepsi, yaitu dengan kata kunci throw. Throw digunakan untuk melempar eksepsi secara manual. Bentuk umum throw adalah sebagai berikut:



```
TEORI > Tugas_7_Exception > 12.2-Throws > Throw1.java > Throw1 > main(String[] args)
1 public class Throw1 {
    Run main | Debug main
2     public static void main(String[] args){
3         RuntimeException x;
4         x= new RuntimeException("eksepsi RuntimeException");
5
6         System.out.println("sebelum throw");
7         throw(x);
8     }
9 }
10

PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI-OBJECT\GIT\Object-Oriented-Programming> java Throw1
sebelum throw
Exception in thread "main" java.lang.RuntimeException: eksepsi RuntimeException
    at Throw1.main(Throw1.java:4)
PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMROGRAMAN-BERORIENTASI-OBJECT\GIT\Object-Oriented-Programming>
```

12.3 throws

Kata kunci throws berfungsi untuk memberitahukan bahwa suatu metode ada kemungkinan menghasilkan suatu eksepsi. Bentuk umum dari throws adalah sebagai berikut:

```
tipe namaMethod(daftarParameter) throws tipeEksepsi1,
tipeEksepsi2, ...
{ //badan method }
```

Berikut contoh program dalam penggunaan throws:


```
readme.md Throws1.java 1 Finally.java 1, U x
TEORI > Tugas_7_Exception > 12.4-Finally > Finally.java > Finally
1 class Finally
2 {
3     private static int j=0;
4
5     Run main | Debug main
6     public static void main(String[] args)
7     {
8         while(true)
9         {
10             try
11             {
12                 System.out.print("pada saat i = " + j + ": ");
13                 if(j++ == 0)
14                     throw new Exception();
15                 System.out.println("tidak terjadi eksepsi");
16             }
17             catch(Exception x)
18             {
19                 System.out.println("terdapat eksepsi");
20             }
21             finally
22             {
23                 System.out.println("statemen dalam blok finally");
24                 if(j == 3)
25                 {
26                     break;
27                 }
28             }
29         }
30     }
```

INFORMATIKA\PEMROGRAMAN-BERORIENTASI-OBJEK

if (\$?) { java Finally }

pada saat i = 0: terdapat eksepsi

statemen dalam blok finally

pada saat i = 1: tidak terjadi eksepsi

statemen dalam blok finally

pada saat i = 2: tidak terjadi eksepsi

statemen dalam blok finally

PS D:\SEMESTER 3 TEKNIK INFORMATIKA\PEMRO