

Modul Praktikum 6 PBO

Exception Handling

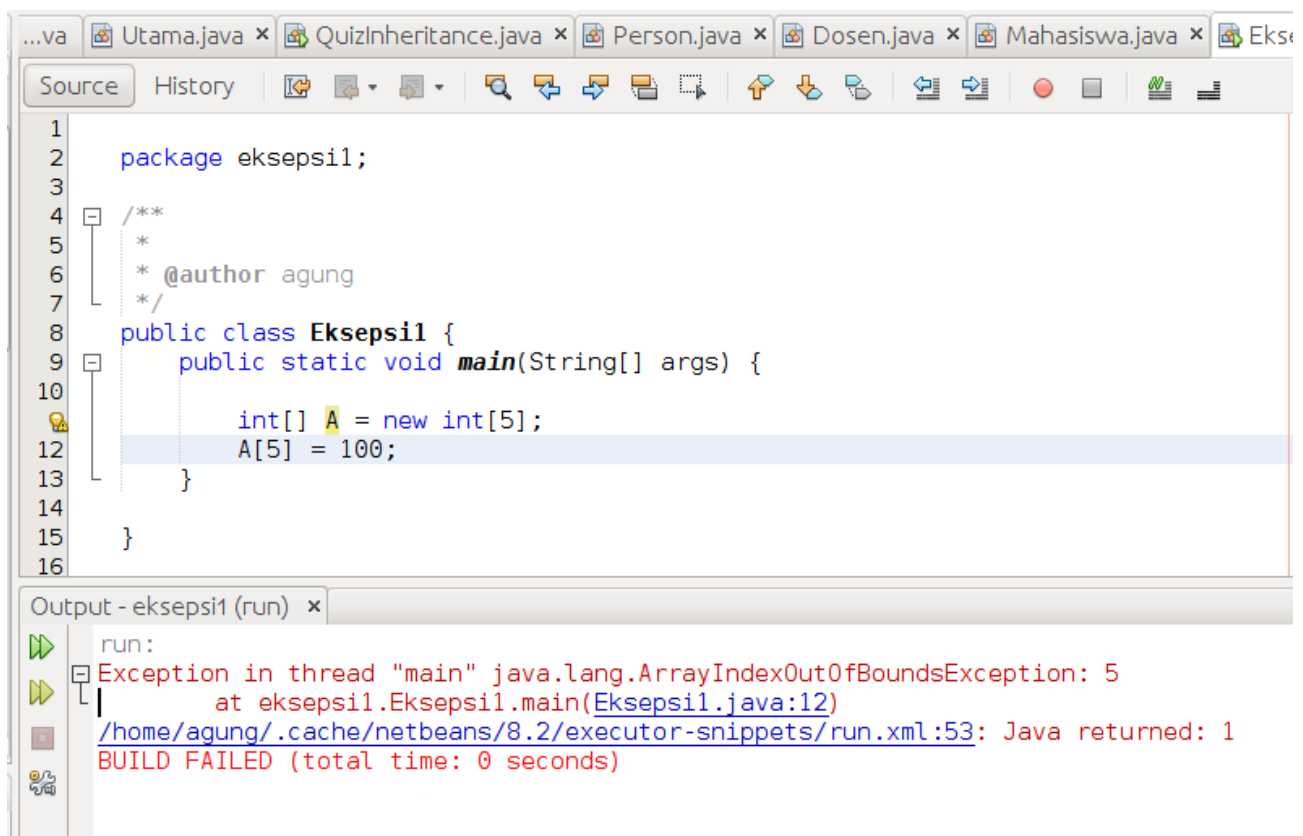
Hal terburuk dan paling dihindari dalam dunia pemrograman adalah **Bug** pada saat proses eksekusi di dalam menjalankan sebuah program. Kesalahan ini disebut dengan **Eksepsi**.

Eksepsi adalah Suatu Kondisi abnormal yang terjadi pada saat runtime.

Pada kebanyakan bahasa pemrograman, saat terjadi kesalahan harus diperiksa secara manual, dan ini memiliki kelemahan karena proses pencarian kelemahan akan sulit dilakukan. Untuk mengatasi permasalahan tersebut, java menyediakan sebuah mekanisme penanganan kesalahan yang disebut dengan **Exception Handling**. Dimana, setiap kesalahan akan dibentuk menjadi sebuah object.

Terdapat lima buah kata kunci yang disediakan oleh java untuk menangani eksepsi, yaitu *try*, *catch*, *throw*, *throws*, and *finally*.

Contoh 1



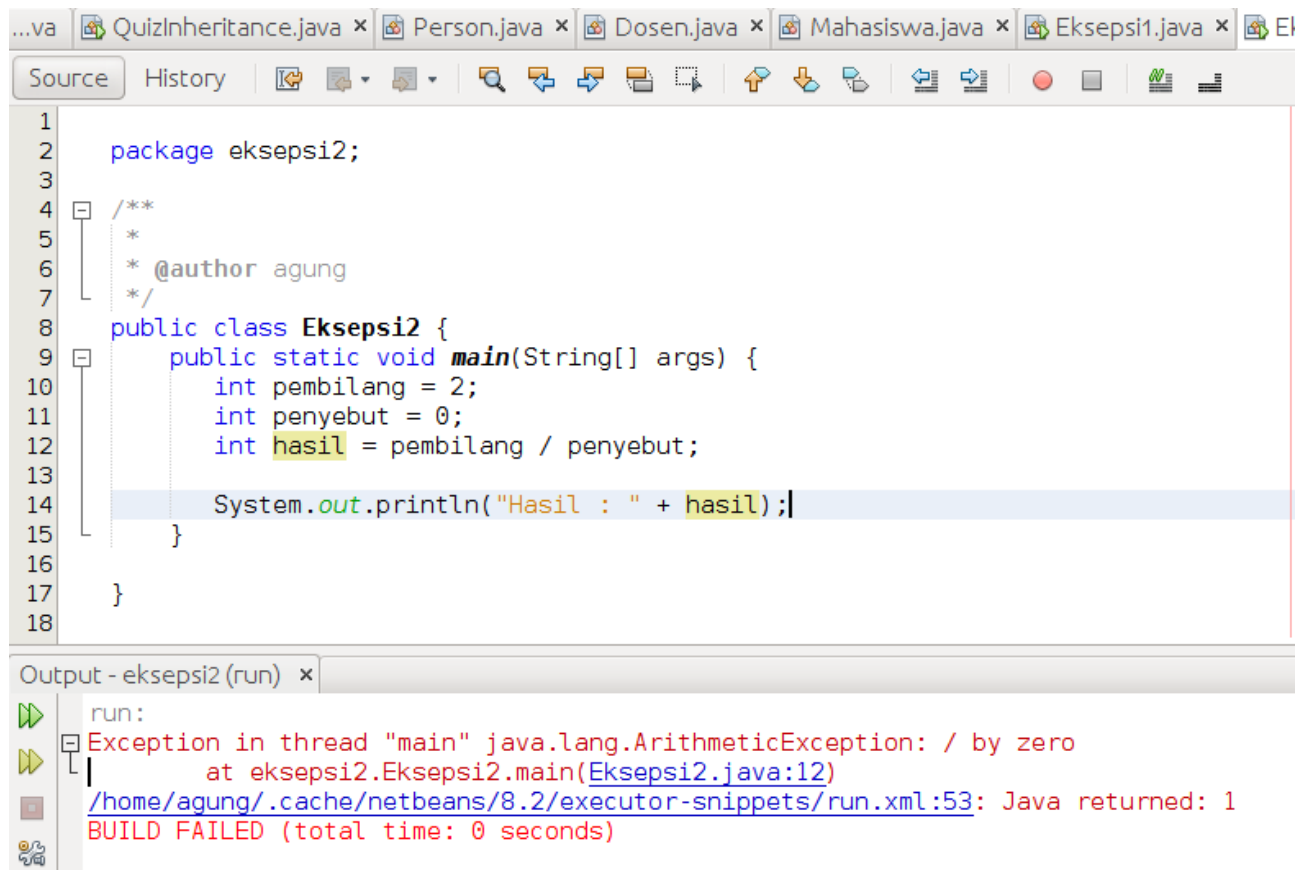
The screenshot shows the NetBeans IDE with a project named 'Eksepsi1'. The 'Source' tab is active, displaying the following Java code:

```
1 package eksepsi1;
2
3
4 /**
5  *
6  * @author agung
7  */
8 public class Eksepsi1 {
9     public static void main(String[] args) {
10
11         int[] A = new int[5];
12         A[5] = 100;
13     }
14 }
15
16
```

The 'Output - eksepsi1 (run)' tab is also visible, showing the following error message:

```
run:
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
    at eksepsi1.Eksepsi1.main(Eksepsi1.java:12)
/home/agung/.cache/netbeans/8.2/executor-snippets/run.xml:53: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```

contoh 2



The screenshot shows an IDE with several open files: QuizInheritance.java, Person.java, Dosen.java, Mahasiswa.java, Eksepsi1.java, and Eksepsi2.java. The 'Source' tab is active, displaying the code for Eksepsi2.java. The code is as follows:

```
1 package eksepsi2;
2
3
4 /**
5  *
6  * @author agung
7  */
8 public class Eksepsi2 {
9     public static void main(String[] args) {
10         int pembilang = 2;
11         int penyebut = 0;
12         int hasil = pembilang / penyebut;
13
14         System.out.println("Hasil : " + hasil);
15     }
16 }
17
18
```

Below the code editor, the 'Output - eksepsi2 (run)' window shows the following error message:

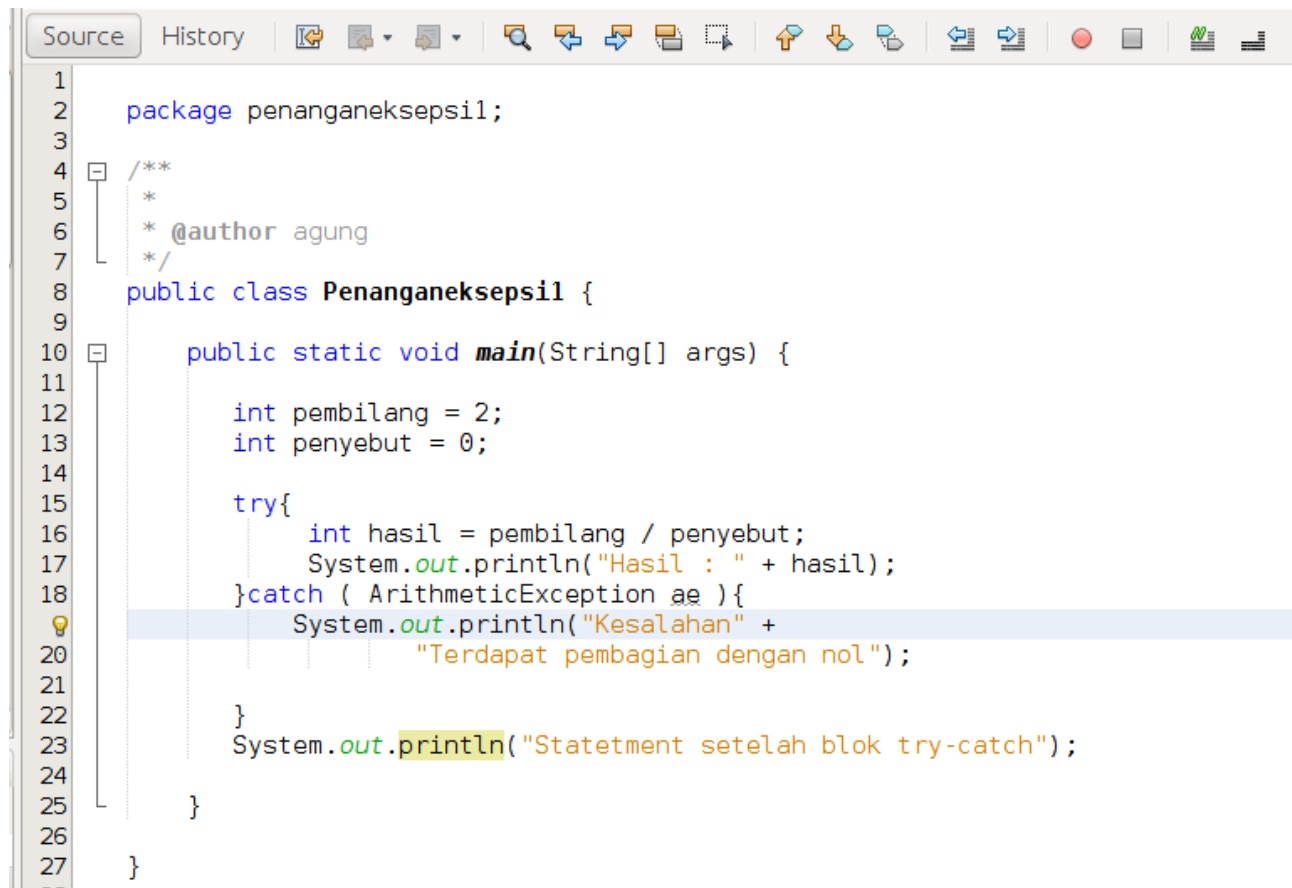
```
run:
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at eksepsi2.Eksepsi2.main(Eksepsi2.java:12)
/home/agung/.cache/netbeans/8.2/executor-snippets/run.xml:53: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```

Type Eksepsi dalam Java

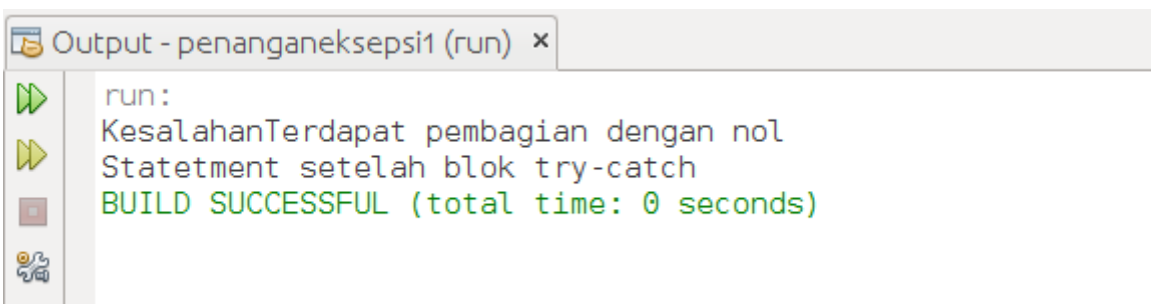
Exception Class Hierarchy		
Throwable	Error	LinkageError, ... VirtualMachineError, ...
	Exception	ClassNotFoundException, CloneNotSupportedException, IllegalAccessException, InstantiationException, InterruptedException, IOException, EOFException, FileNotFoundException, ... RuntimeException, ArithmeticException, ArrayStoreException, ClassCastException, IllegalArgumentException, (IllegalThreadStateException and NumberFormatException as subclasses) IllegalMonitorStateException, IndexOutOfBoundsException, NegativeArraySizeException, NullPointerException, SecurityException ...

Exception Handling

1. Menggunakan kata kunci try dan catch



```
1
2 package penanganeksepsi1;
3
4 /**
5  *
6  * @author agung
7  */
8 public class Penanganeksepsi1 {
9
10     public static void main(String[] args) {
11
12         int pembilang = 2;
13         int penyebut = 0;
14
15         try{
16             int hasil = pembilang / penyebut;
17             System.out.println("Hasil : " + hasil);
18         }catch ( ArithmeticException ae ){
19             System.out.println("Kesalahan" +
20                 "Terdapat pembagian dengan nol");
21
22         }
23         System.out.println("Statetment setelah blok try-catch");
24     }
25 }
26
27 }
```

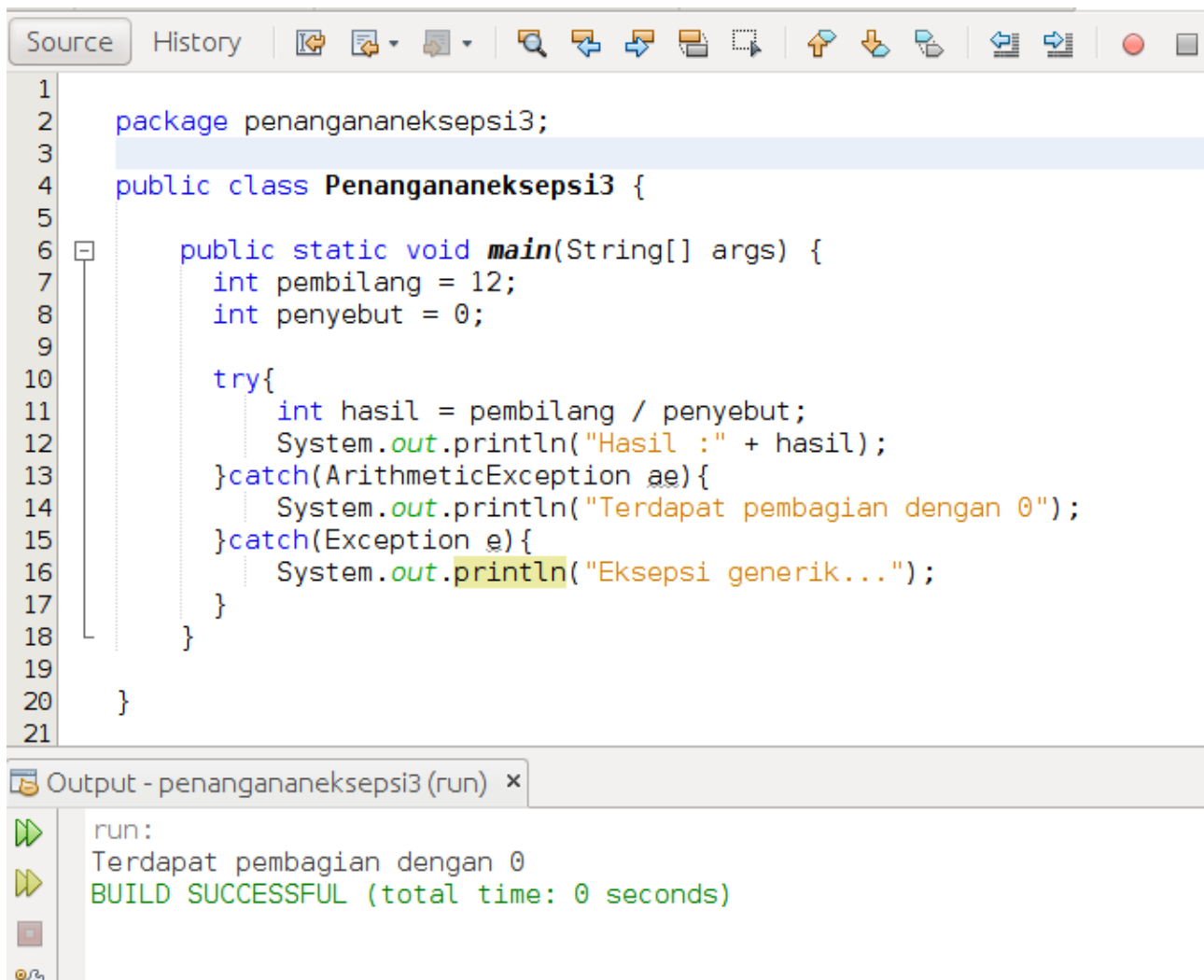


```
Output - penanganeksepsi1 (run) x
run:
KesalahanTerdapat pembagian dengan nol
Statetment setelah blok try-catch
BUILD SUCCESSFUL (total time: 0 seconds)
```

2.

```
Source History
1 package penangananeksepsi2;
2
3
4 /**
5  *
6  * @author agung
7  */
8 public class Penangananeksepsi2 {
9
10     public static void main(String[] args) {
11         int[] A = new int[5];
12
13         try{
14             for (int i = 0; i < 5; i++){
15                 A[i] = i * 10;
16             }
17             System.out.println("Nilai pada indeks ke-9 : " + A[9]);
18         }catch (ArrayIndexOutOfBoundsException e){
19             System.out.println("Tidak terdapat indeks ke " + e.getMessage());
20         }
21         for (int i = 0; i < 5; i++){
22             System.out.println("Elemen ke- " + i + ":" + A[i]);
23         }
24     }
25 }
26
27
28
```

```
Output - penangananeksepsi2 (run) x
run:
Tidak terdapat indeks ke 9
Elemen ke- 0:0
Elemen ke- 1:10
Elemen ke- 2:20
Elemen ke- 3:30
Elemen ke- 4:40
BUILD SUCCESSFUL (total time: 0 seconds)
```



The screenshot shows an IDE window with a source code editor and an output console. The source code defines a package `penangananeksepsi3` and a class `Penangananeksepsi3` with a `main` method. The `main` method initializes `pembilang` to 12 and `penyebut` to 0, then attempts a division within a `try` block. It catches `ArithmeticException` and prints "Terdapat pembagian dengan 0", and catches a generic `Exception` and prints "Eksepsi generik...". The output console shows the execution result, confirming the division by zero message and a successful build.

```
1
2 package penangananeksepsi3;
3
4 public class Penangananeksepsi3 {
5
6     public static void main(String[] args) {
7         int pembilang = 12;
8         int penyebut = 0;
9
10        try{
11            int hasil = pembilang / penyebut;
12            System.out.println("Hasil :" + hasil);
13        }catch(ArithmeticException ae){
14            System.out.println("Terdapat pembagian dengan 0");
15        }catch(Exception e){
16            System.out.println("Eksepsi generik...");
17        }
18    }
19
20 }
21
```

Output - penangananeksepsi3 (run) x

```
run:
Terdapat pembagian dengan 0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Tugas

Cari contoh Type Exception yang lain, dan masukkan ke dalam laporan praktikum + cantumkan sumber nya. Terima Kasih (Minimal 5)