

MODUL PRAKTIKUM EXCEPTION HANDLING
PEMROGRAMAN BERORIENTASI OBYEK



UIN SUNAN AMPEL
S U R A B A Y A

DOSEN :

Dwi Rolliawati

DISUSUN OLEH :

M.Khotibul Umam (09020620031)

UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
FAKULTAS SAINT DAN TEKNOLOGI
PRODI SISTEM INFORMASI
2020/2021

The screenshot shows the Apache NetBeans IDE interface. The main editor window displays the code for `Exceptionn.java` in the `Percobaan1` package. The code is as follows:

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package Percobaan1;
7
8  /**
9   *
10   * @author NAM
11   */
12  public class Exceptionn {
13
14      public static void main(String[] args) {
15          int a[] = new int[5];
16          a[5] = 100;
17      }
18
19      //public static void main(String[] args) {
20      //    int a [] = new int[5];
21  }
```

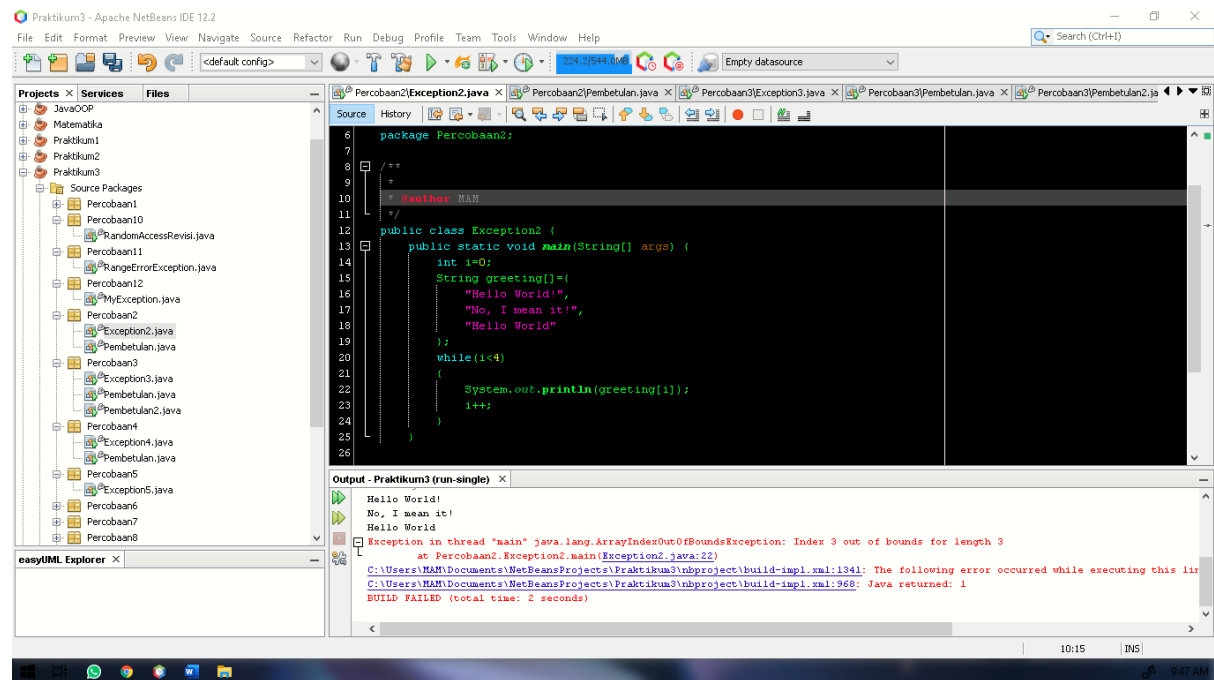
The 'Output' window shows the error message:

```
Output - Praktikum3 (run-single) x
C:\Users\NAM>java -jar
Updating property file: C:\Users\NAM\Documents\NetBeansProjects\Praktikum3\build\build-jar.properties
Compiling 1 source file to C:\Users\NAM\Documents\NetBeansProjects\Praktikum3\build\classes
warning: [options] bootstrap class path not set in conjunction with -source 8
1 warning
compile-single:
run-single:
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5
at Percobaan1.Exceptionn.main(Exceptionn.java:16)
C:\Users\NAM\Documents\NetBeansProjects\Praktikum3\build\build-impl.xml:134: The following error occurred while executing this line:
C:\Users\NAM\Documents\NetBeansProjects\Praktikum3\build\build-impl.xml:968: Java returned: 1
BUILD FAILED (total time: 3 seconds)
```

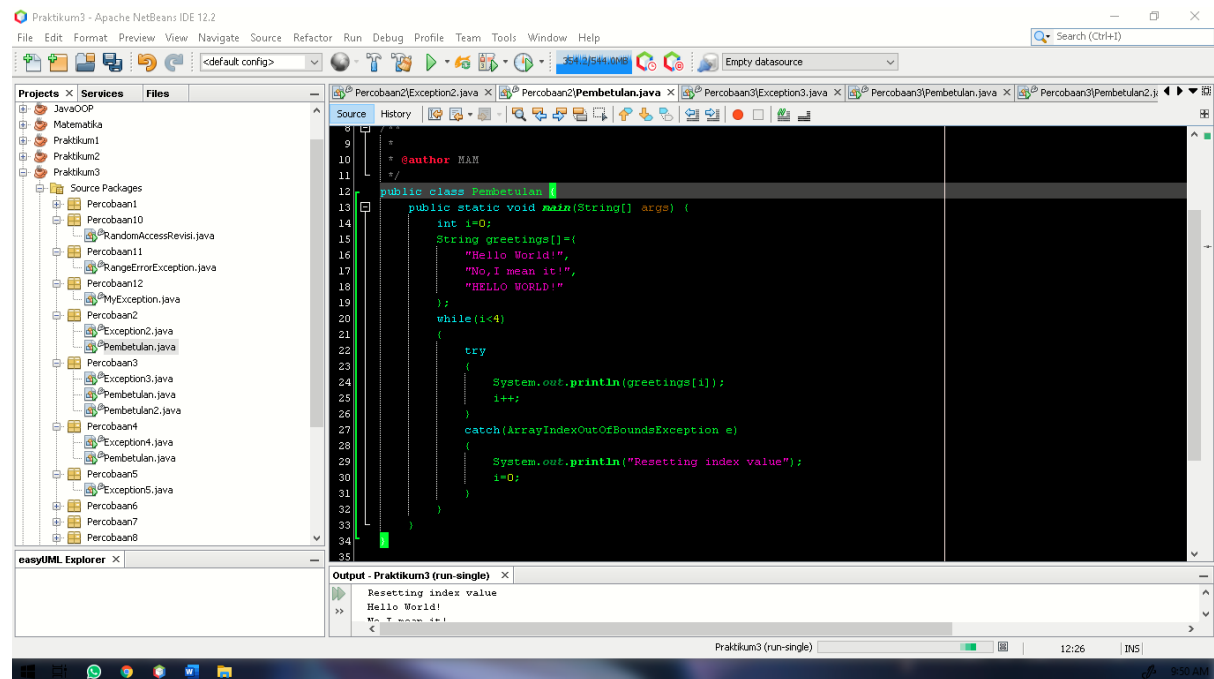
The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, Format, Preview, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar contains icons for file operations and running code. The left sidebar shows the 'Projects' view with a tree structure of source packages and files, including 'Percobaan1' through 'Percobaan6'. The main editor window displays the source code of 'Percobaan1/Exceptionn.java'. The code includes a package declaration, a class declaration, and a main method that throws an exception. The bottom status bar shows the output of a run command: 'run: Terjadi pelanggaran memory BUILD SUCCESSFUL (total time: 2 seconds)'.

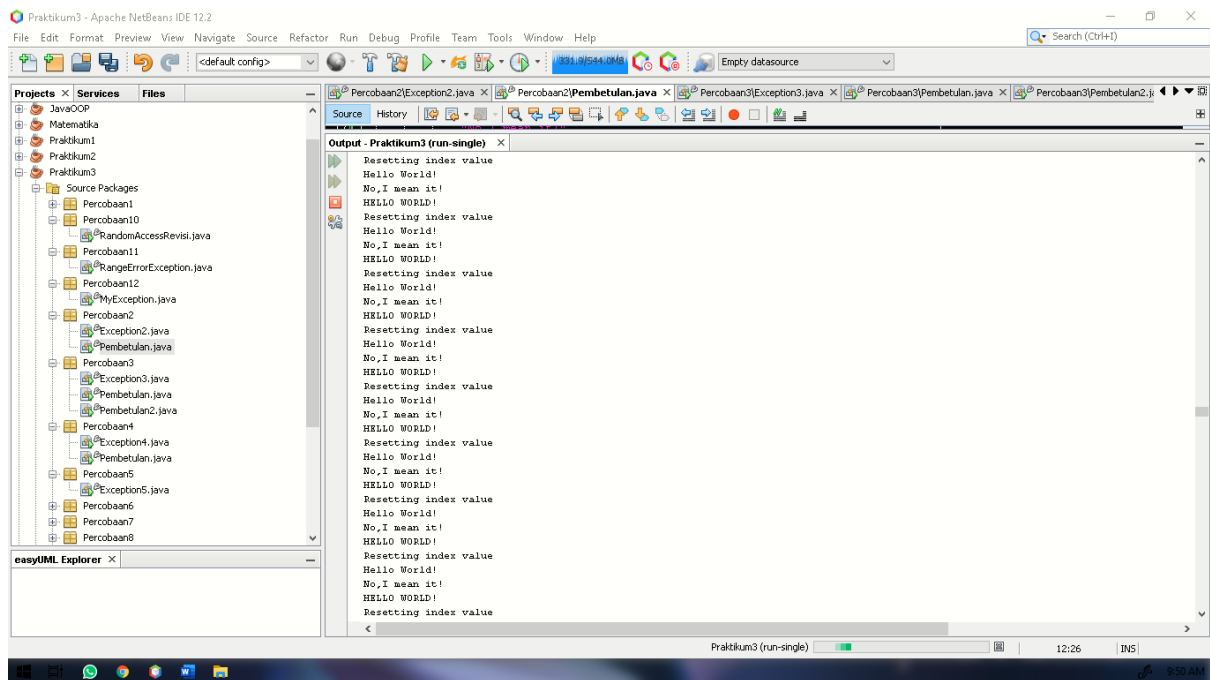
Pada percobaan pertama sebelum pembetulan terjadi `ArrayIndexOutOfBoundsException` dimana itu artinya index nilai array yang diisikan melebihi kapasitas atau diluar dari batas. Pada pembetulan program menggunakan `try - catch` dimana pada `try` nya diisi dengan penyebab exception atau yang berpotensi untuk menimbulkan kesalahan dan pada `catch`nya diisi dengan menampilkan peringatan karena `catch` sendiri berfungsi untuk menangkap kesalahan atau bug yang terjadi dalam block, maka exception handling nya dapat berjalan.

Percobaan 2



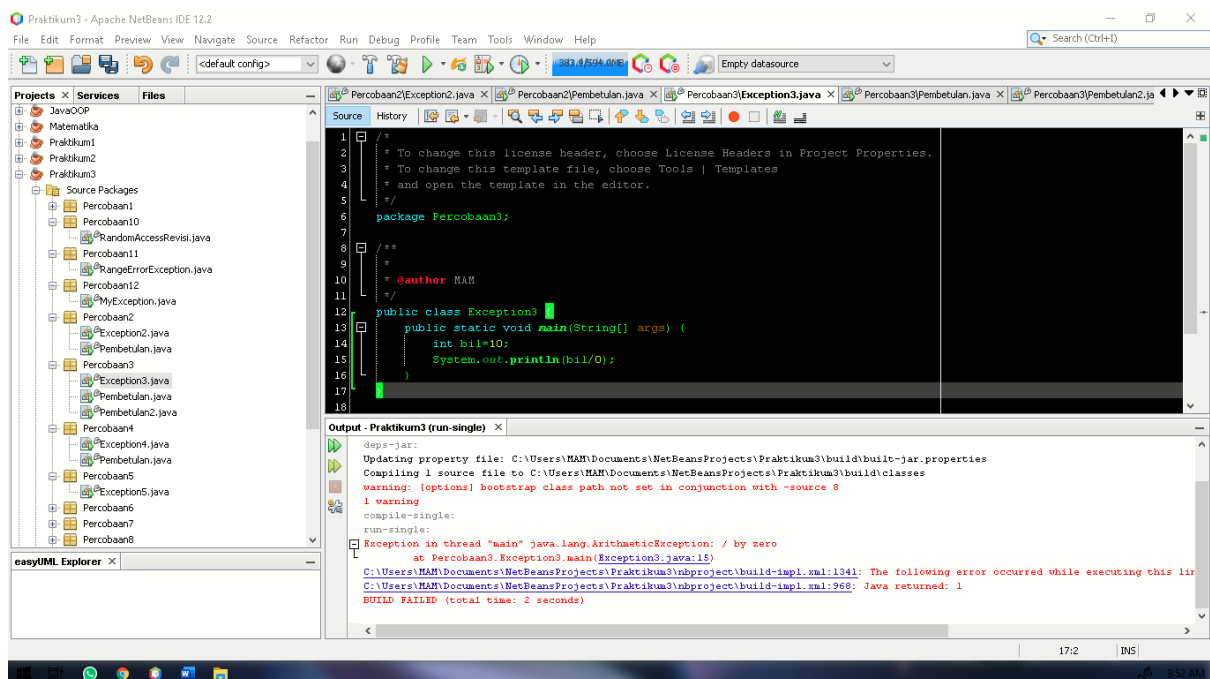
Pembetulan



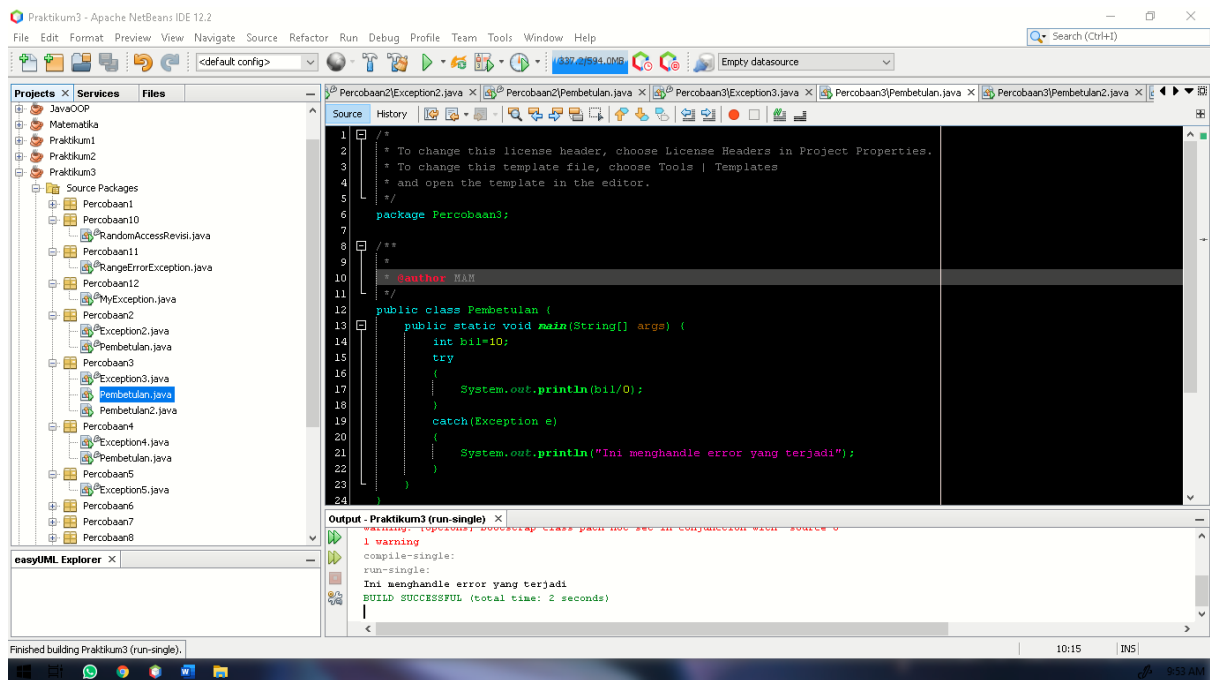


Pada saat pembetulan outputnya di print terus atau infinity looping karena tidak adanya tanda `i++` dalam catchnya sehingga dapat menghentikan looping tersebut jika nilainya sudah sama dengan kondisi yang ditentukan. penyebab kedua adalah seharusnya bukan menggunakan lengan arraynya agar tidak infinity loop.

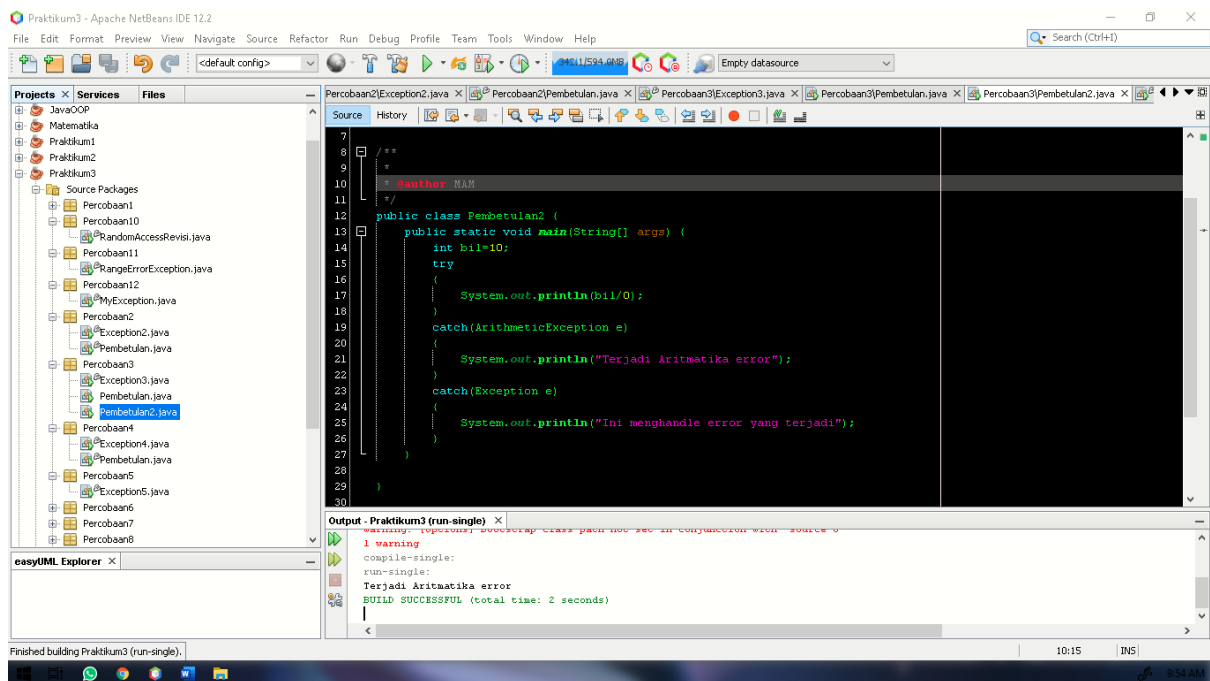
Percobaan 3



Pembetulan

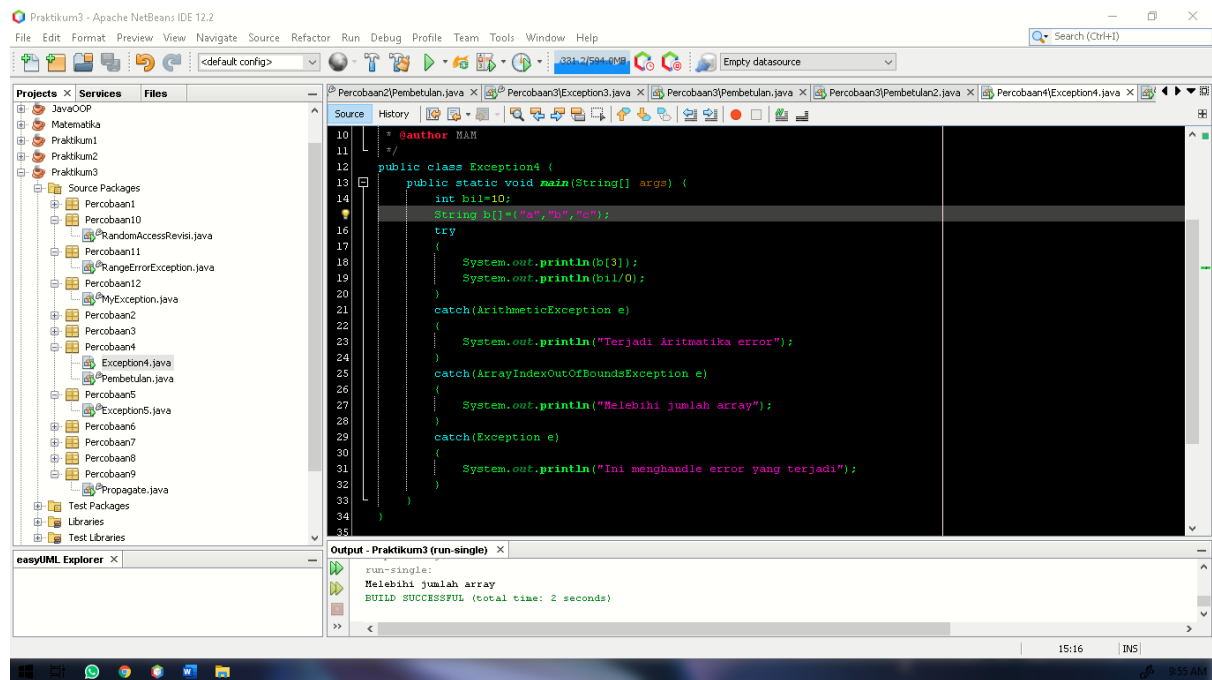


Pembetulan 2

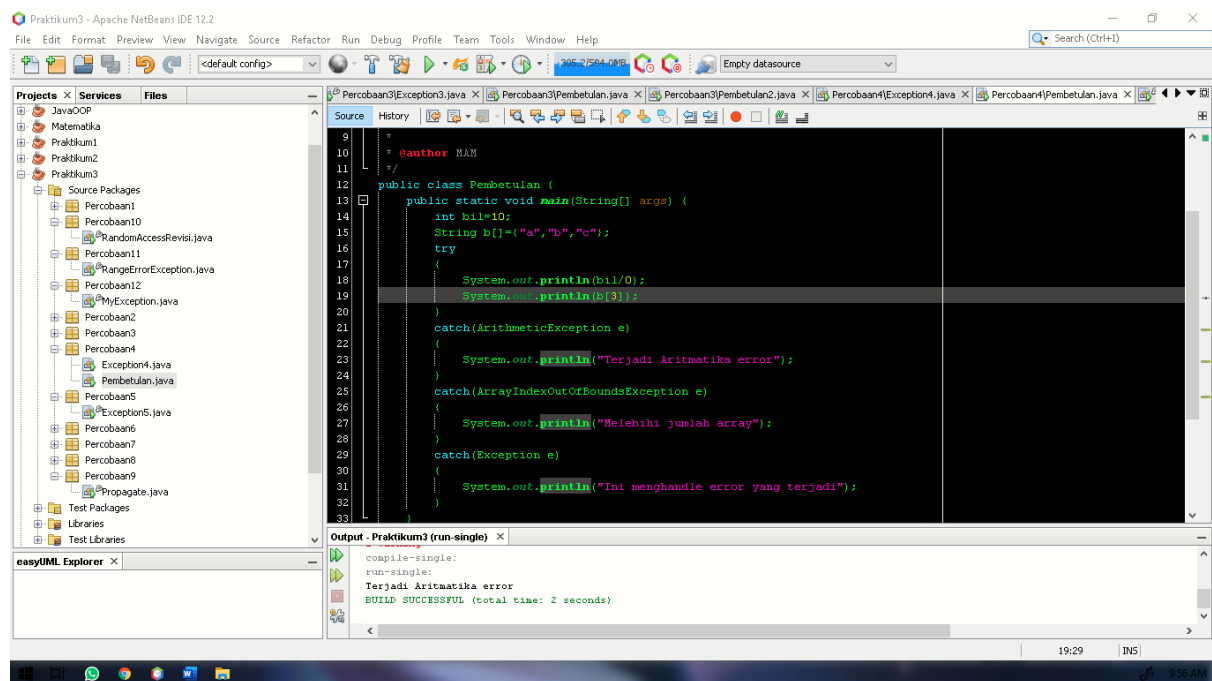


Pada percobaan ketiga ini pembetulan pertama pada catch nya kita memanggil exception untuk menampilkan peringatan tampililah peringatan tersebut, namun untuk uji coba pembetulan berikutnya kita memanggil dua exception sekaligus namun sistem tetap memilih menampilkan ArithmeticException karena lebih spesifik.

Percobaan 4

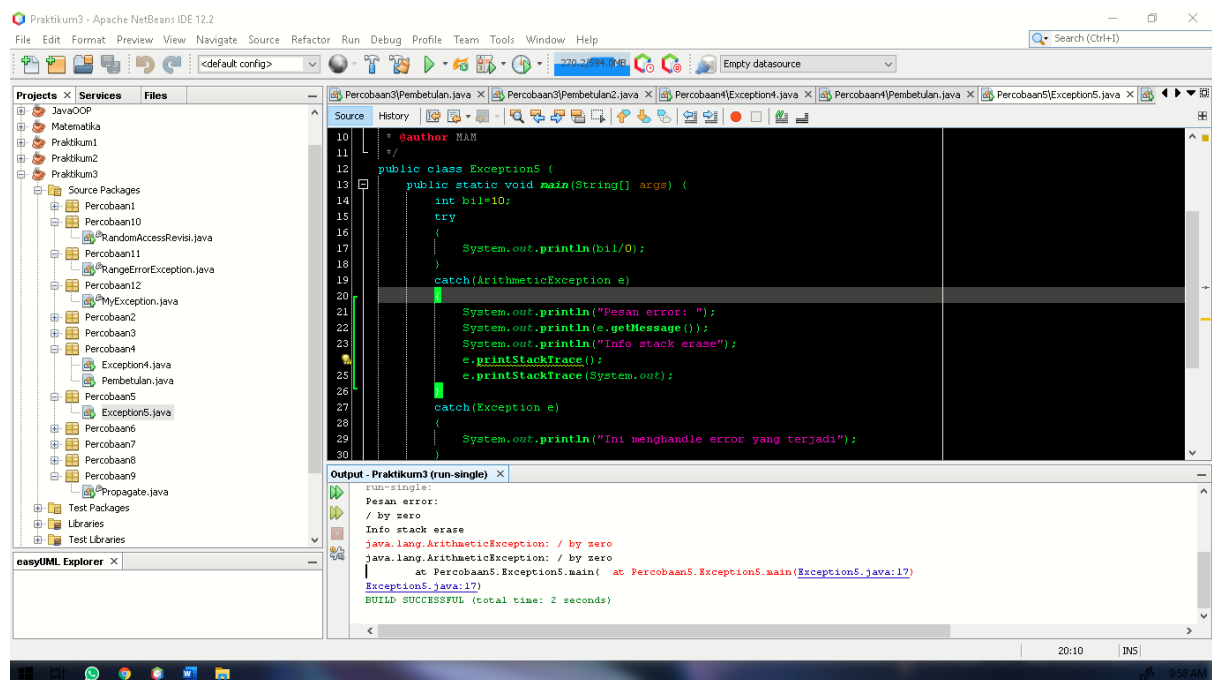


Pembetulan



Pada uji coba 4 ini yang tampil pertama adalah `ArrayIndexOutOfBoundsException` karena pada trynya sistem disuruh untuk menampilkan index array yang di luar batas lalu pada percobaan selanjutnya terjadi kesalahan aritmatik seperti pembagian dengan nol, maka tampililah `catch ArithmeticException`.

Percobaan 5



The screenshot shows the Apache NetBeans IDE interface. On the left, the 'Projects' pane displays a tree structure of files and folders, including 'Percobaan1' through 'Percobaan10' and 'Percobaan11' through 'Percobaan12'. The main editor window shows the source code of 'Exception5.java'. The code defines a public class 'Exception5' with a static method 'main' that takes a 'String[] args' parameter. Inside the 'main' method, a variable 'bil' is initialized to 10, and a 'try' block is used to execute 'System.out.println(bil/0);'. This is followed by a 'catch' block for 'ArithmeticException e', which prints a message, the exception's message, and the stack trace. A second 'catch' block for 'Exception e' prints a message indicating that the error is being handled. The 'Output' pane at the bottom shows the results of running the program, displaying the error message and stack trace for the 'ArithmeticException'.

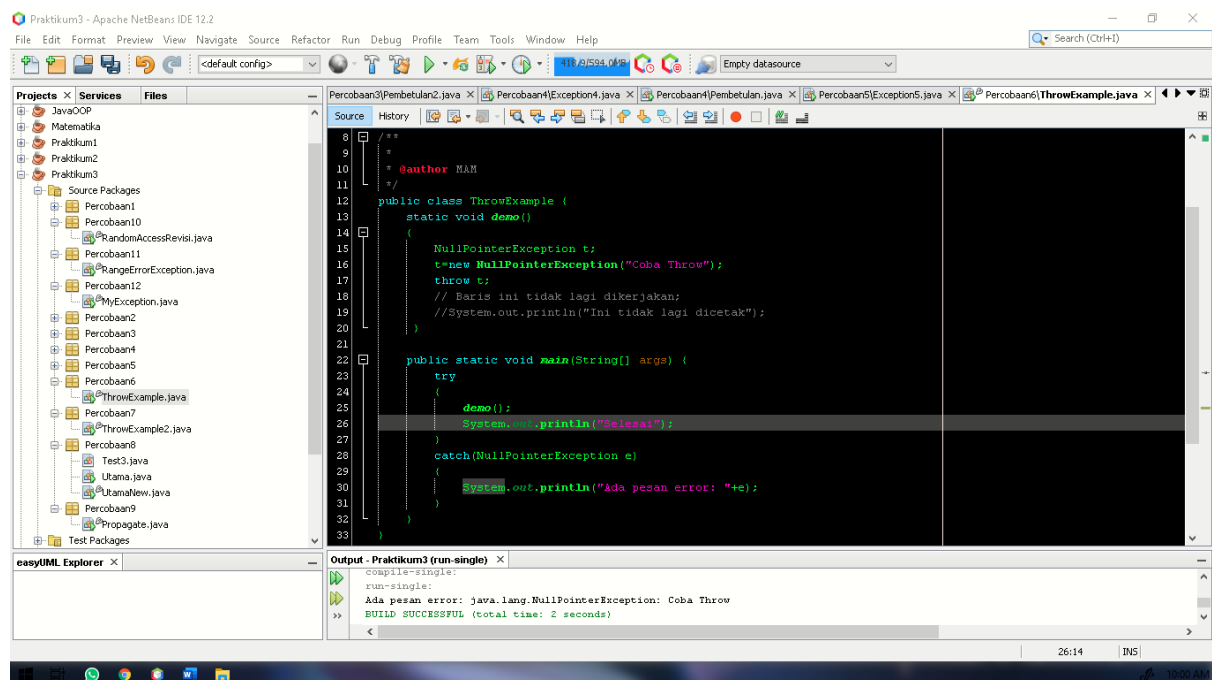
```
10  * @author MAM
11  */
12  public class Exception5 {
13      public static void main(String[] args) {
14          int bil=10;
15          try
16          {
17              System.out.println(bil/0);
18          }
19          catch(ArithmeticException e)
20          {
21              System.out.println("Pesan error: ");
22              System.out.println(e.getMessage());
23              System.out.println("Info stack erase");
24              e.printStackTrace();
25              e.printStackTrace(System.out);
26          }
27          catch(Exception e)
28          {
29              System.out.println("Ini menghandle error yang terjadi");
30          }
31      }
32  }
```

Output - Praktikum3 (run-single) X

```
run-single:
Pesan error:
/ by zero
Info stack erase
java.lang.ArithmeticException: / by zero
at java.lang.ArithmeticException: / by zero
at Percobaan5.Exception5.main( at Percobaan5.Exception5.main(Exception5.java:17)
Exception5.java:17)
BUILD SUCCESSFUL (total time: 2 seconds)
```

Dalam percobaan 5 menggunakan konsep exception “try... catch”, dimana yang dimaksud ialah menangani munculnya error pada proses. Konsep “catch” berfungsi untuk menangani jenis exception, kemudian kesalahan dianggap sebagai objek. Pada “catch” yang ke-2 tidak muncul outputnya dikarenakan exception yang muncul sudah terlebih dahulu ditangani oleh “catch” yang pertama.

Percobaan 6



The screenshot shows the Apache NetBeans IDE interface. On the left, the 'Projects' pane displays a tree structure of files and folders, including 'Percobaan1' through 'Percobaan10' and 'Percobaan11' through 'Percobaan12'. The main editor window shows the source code of 'ThrowExample.java'. The code defines a public class 'ThrowExample' with a static method 'demo' that throws a 'NullPointerException'. The 'main' method calls 'demo()' and prints a message. A 'catch' block for 'NullPointerException e' prints a message indicating that the error is being handled. The 'Output' pane at the bottom shows the results of running the program, displaying the error message and stack trace for the 'NullPointerException'.

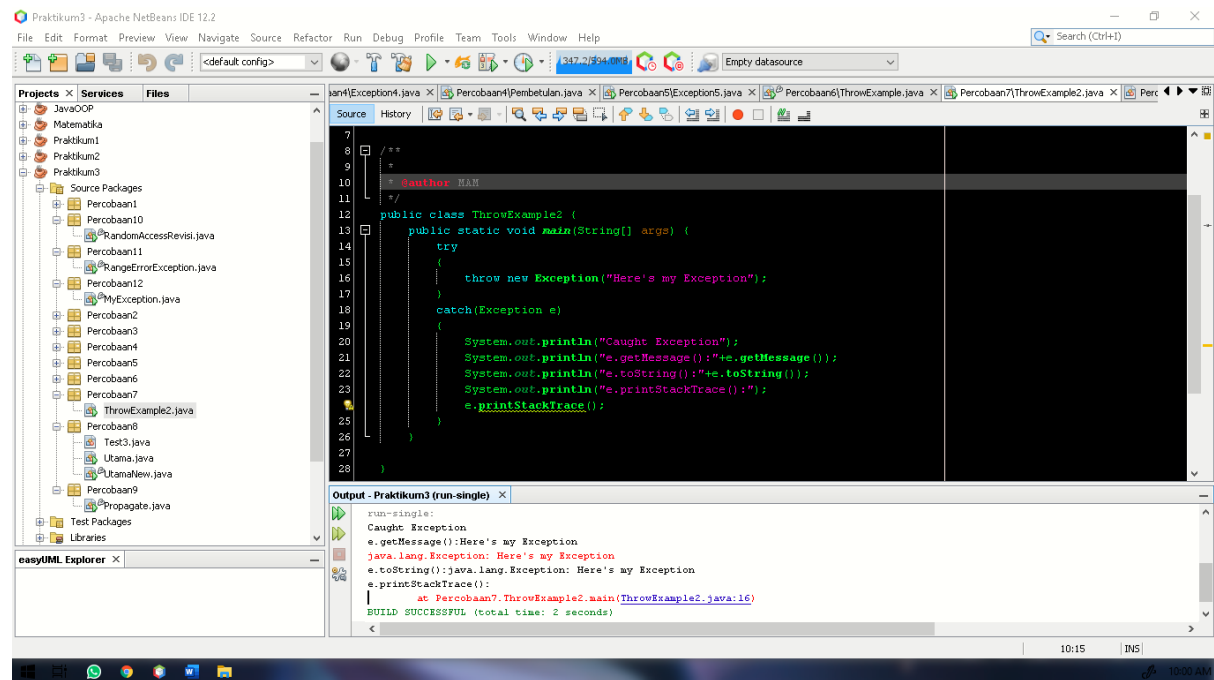
```
8  /**
9   *
10  * @author MAM
11  */
12  public class ThrowExample {
13      static void demo()
14      {
15          NullPointerException t;
16          t=new NullPointerException("Coba Throw");
17          throw t;
18          // Baris ini tidak lagi dikerjakan;
19          //System.out.println("Ini tidak lagi dicetak");
20      }
21
22      public static void main(String[] args) {
23          try
24          {
25              demo();
26              System.out.println("Selesai");
27          }
28          catch(NullPointerException e)
29          {
30              System.out.println("Ada pesan error: "+e);
31          }
32      }
33  }
```

Output - Praktikum3 (run-single) X

```
compile-single:
run-single:
Ada pesan error: java.lang.NullPointerException: Coba Throw
BUILD SUCCESSFUL (total time: 2 seconds)
```

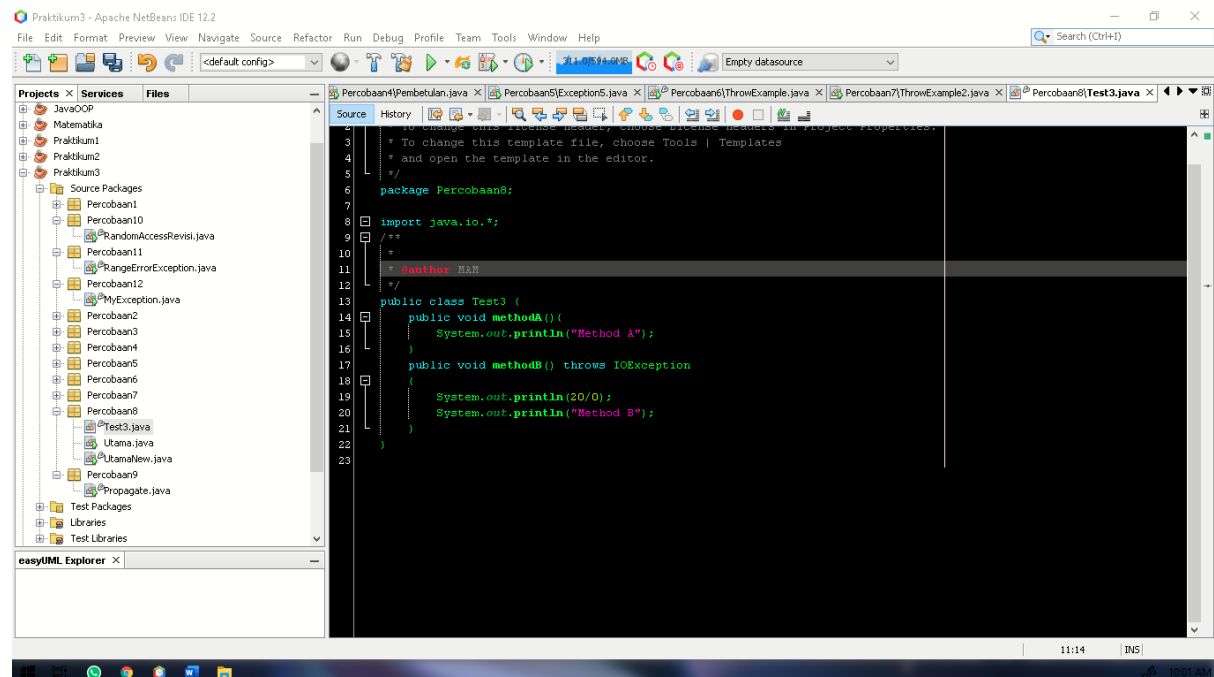
Dalam percobaan 6 konsep “try” digunakan untuk menangani munculnya kesalahan method yang bernama “demo” dalam proses output. “catch” akan menangani exception tersebut lalu akan dianggap menjadi objek dan outputnya

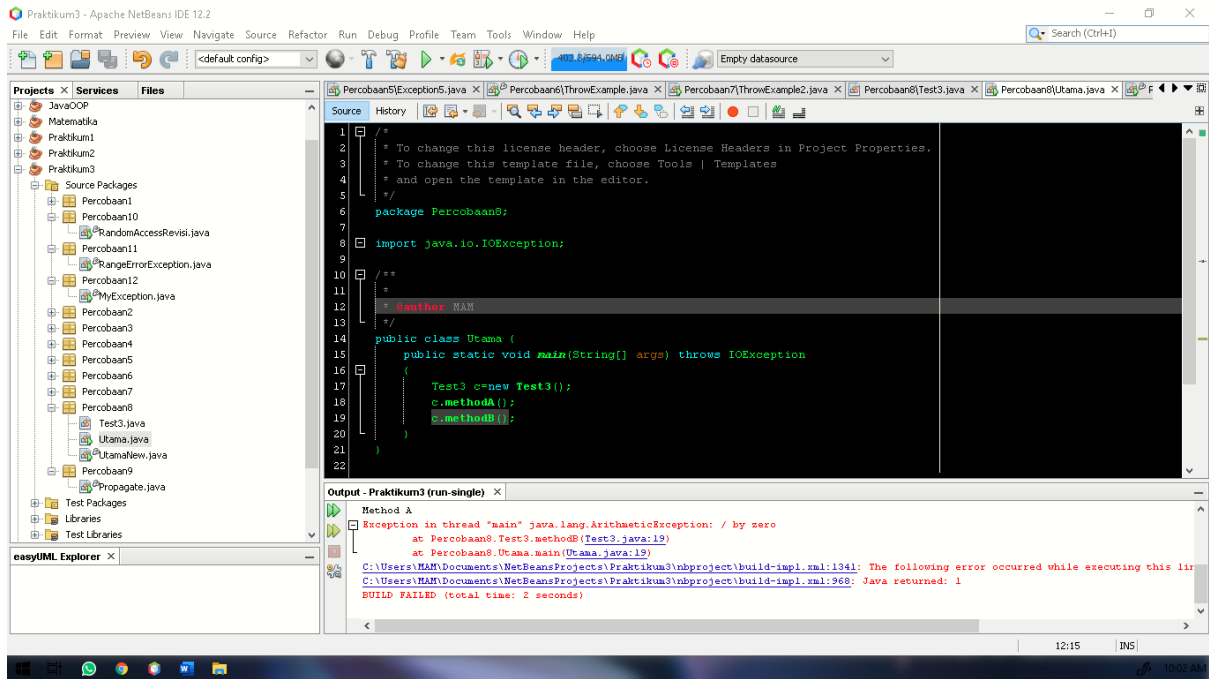
Percobaan 7



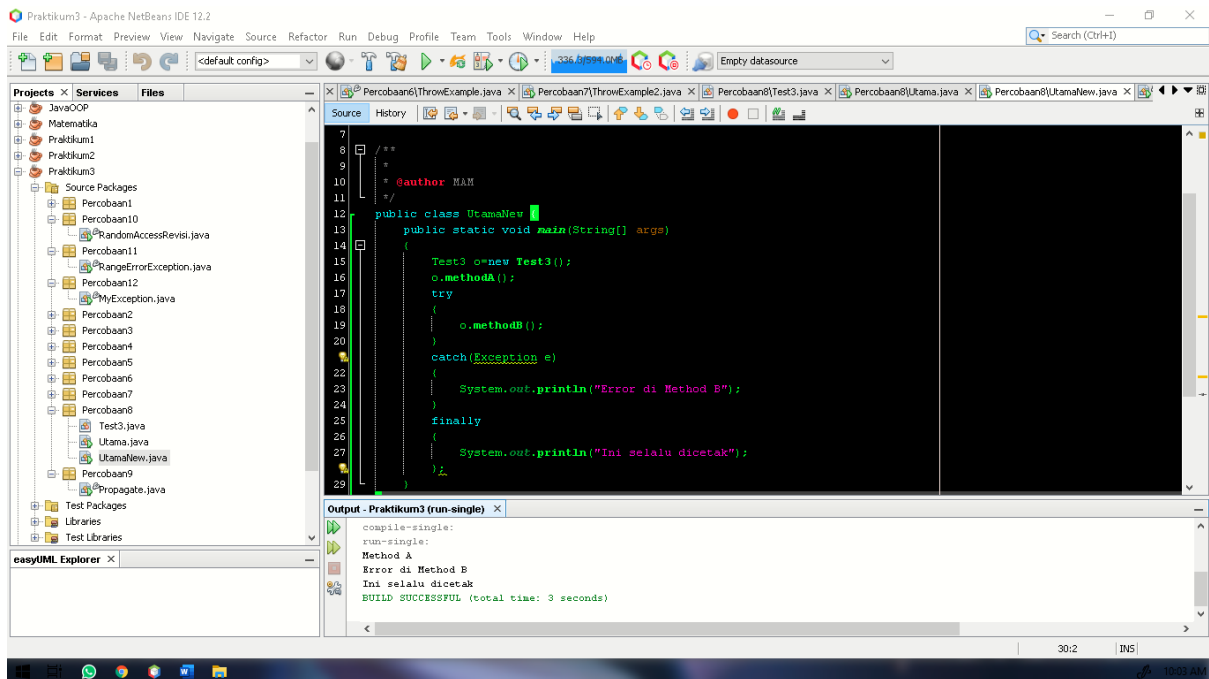
Dalam percobaan 7 “try” ialah pesan yang berawalan perintah “throw new” yang memiliki tipe “exception”. Merupakan subclass dari class exception. Sehingga dianggap suatu kesalahan yang diharuskan ditangani oleh keyword “catch”.

Percobaan 8



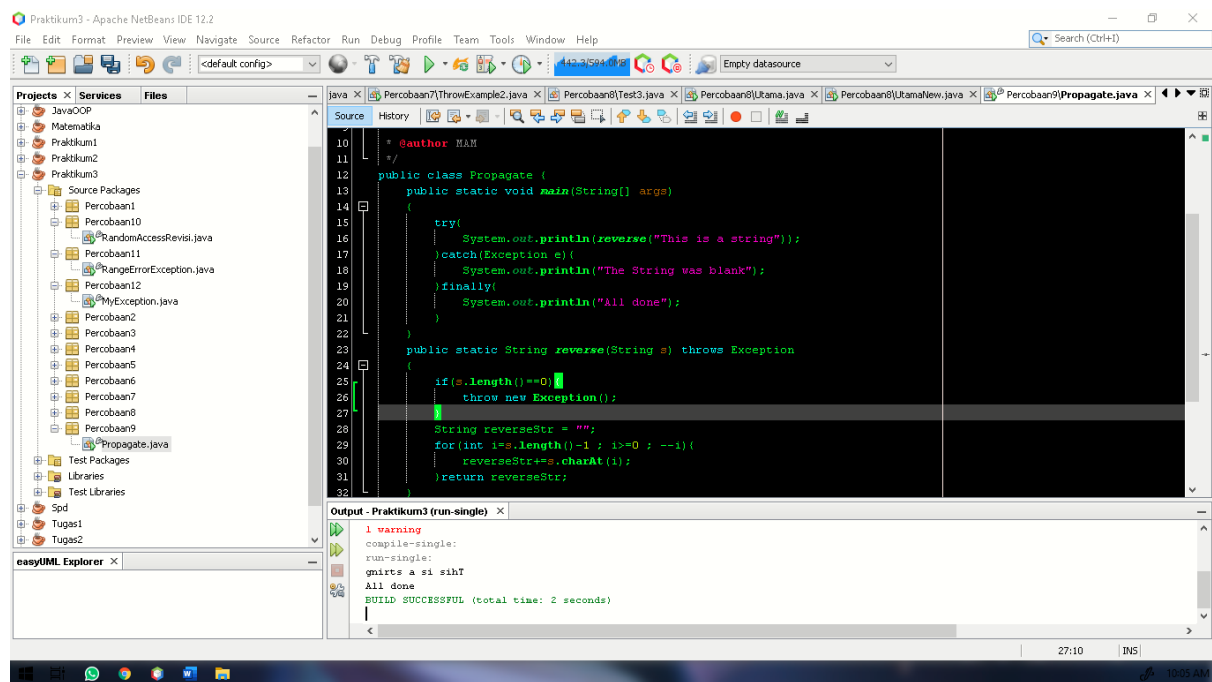


Pembaharuan class utama



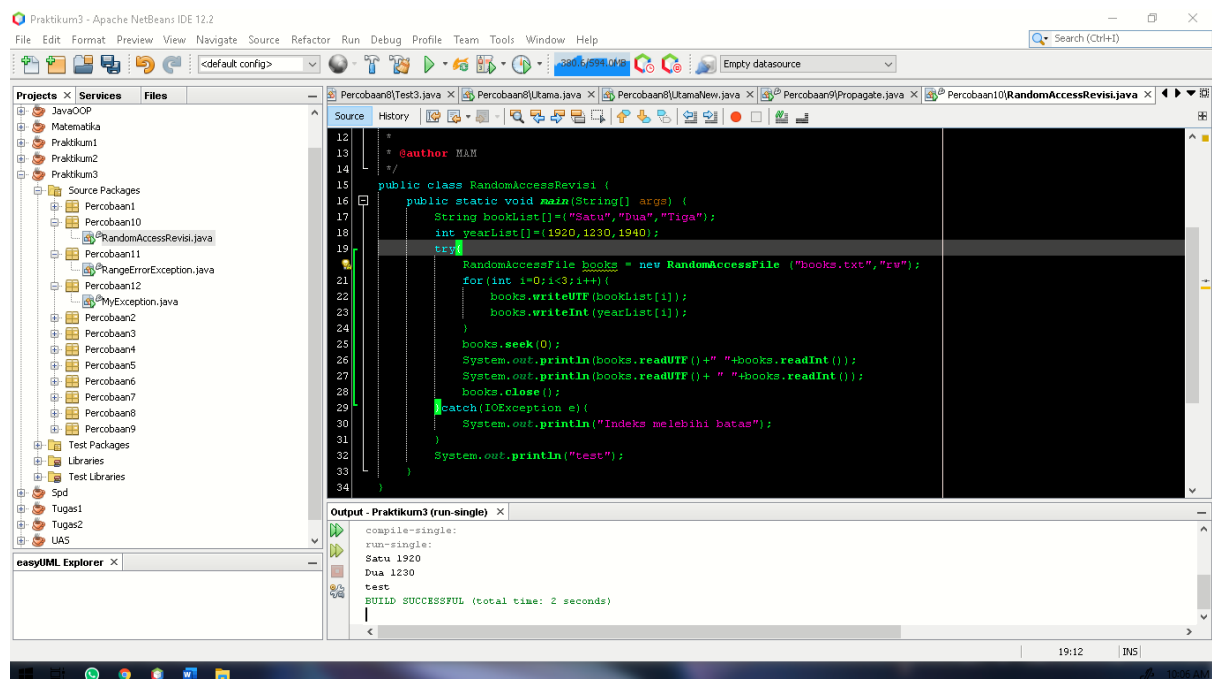
Pada percobaan 8 ini kita mempelajari penggunaan IOException dimana pada Java memiliki fitur yang disebut "pengecualian yang dicentang". Itu berarti ada beberapa jenis pengecualian, yaitu pengecualian Subclass tetapi bukan RuntimeException, sehingga jika suatu metode dapat membuangnya, maka harus mencantumkanannya dalam deklarasi melempar, katakan: void method() throws IOException. Oleh karena itu kita masukan pada try - catch untuk menampilkan error atau kesalahan yang terjadi serta menambahkan finally untuk menampilkan keterangan, karena finally akan selalu di eksekusi meskipun adanya kesalahan atau tidak pada block tersebut.

Percobaan 9



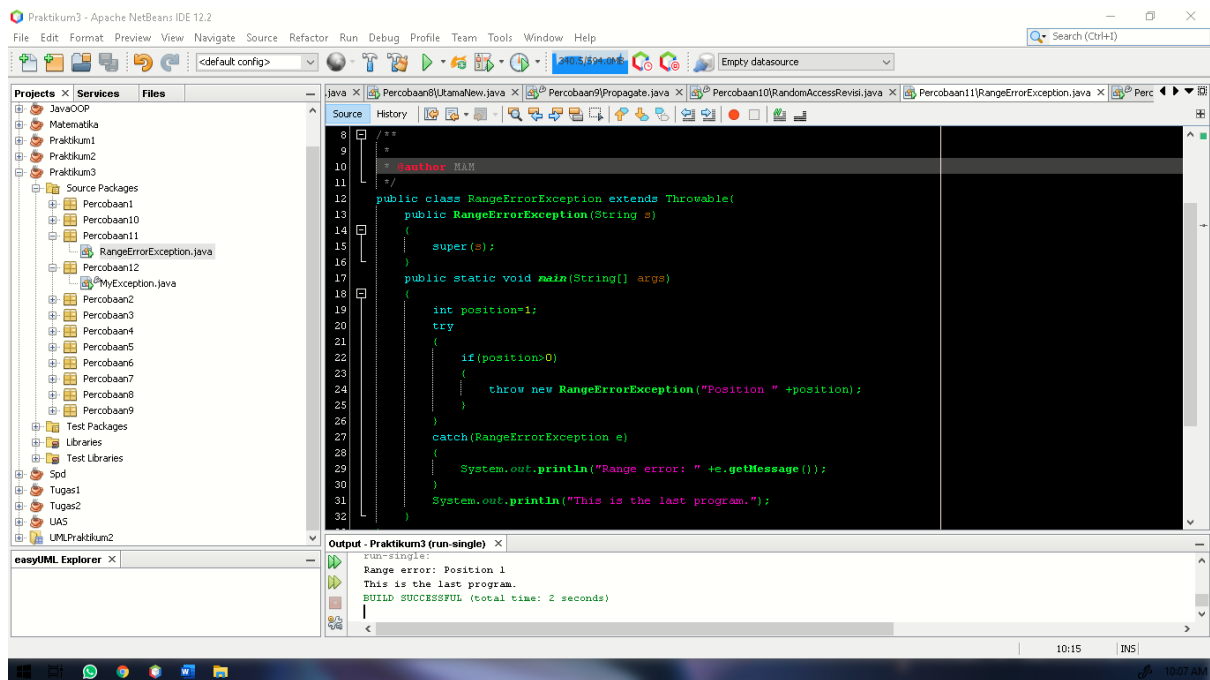
Pada percobaan 9 ini kita diajarkan menggunakan String reverse dan mengkombinasikannya dengan throw, dimana kegunaan dari reverse sendiri adalah untuk membalikkan string yang telah di set pada reverse tersebut dan melemparnya menggunakan throw.

Percobaan 10



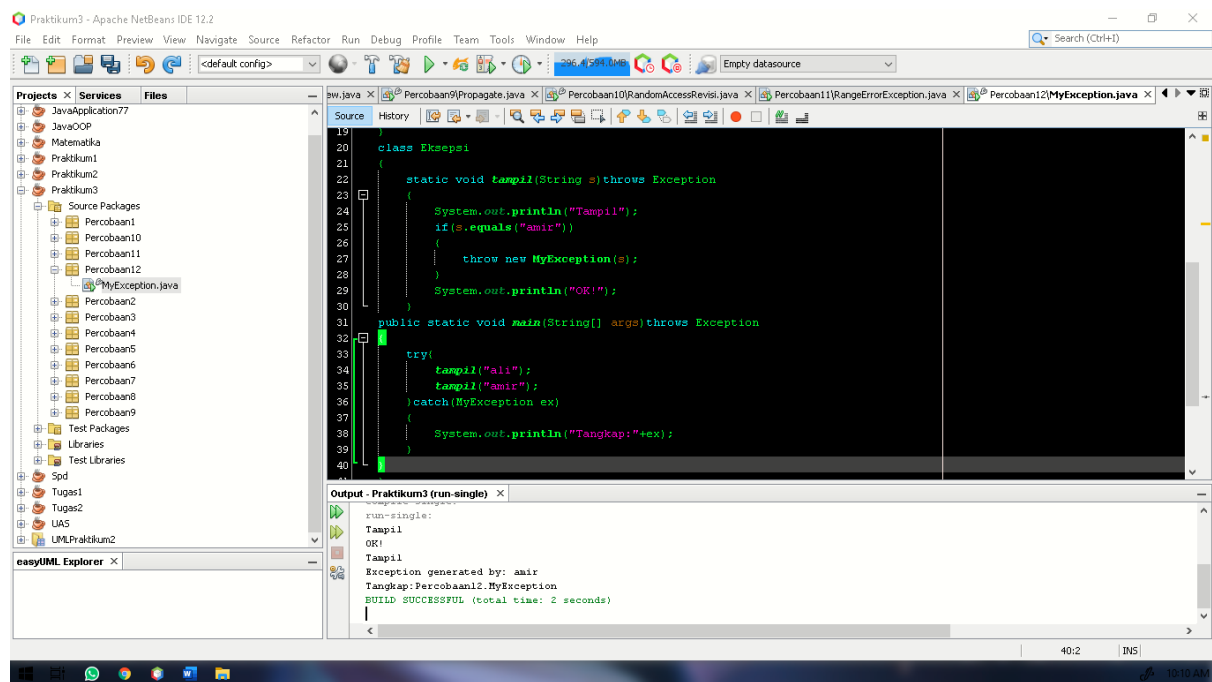
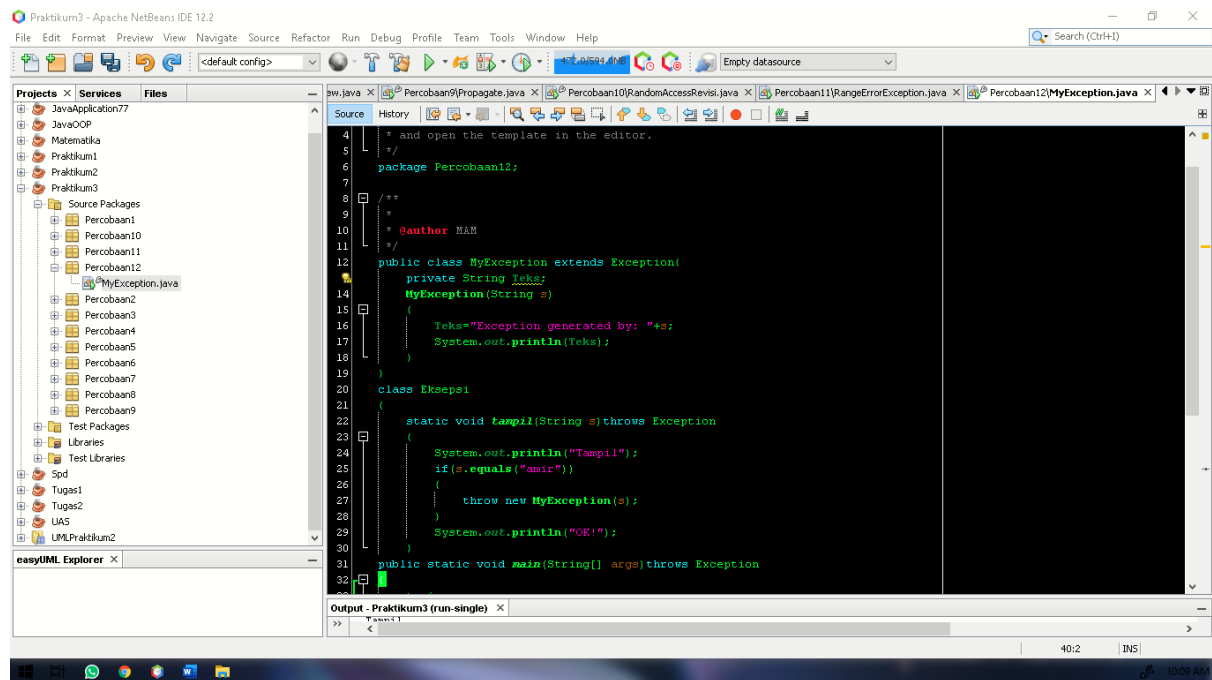
Pada percobaan 10 ini menurut saya lumayan rumit dalam penggunaan IOException karena entah mengapa terdapat error pada deklarasi dari class tetapi setelah saya tambahkan import java.io*; semua jadi berjalan lancar bahkan catch yang sudah dibuat itu tidak menampilkan peringatan yang artinya program berjalan lancar tanpa adanya error.

Percobaan 11



Pada percobaan 11 ini kita diajarkan mengenai RangeErrorException dimana ini menunjukkan terjadinya kesalahan saat nilai tidak dalam set atau rentang nilai yang diizinkan, oleh karena itu RangeError ini dilemparkan saat meneruskan nilai sebagai argumen ke fungsi atau method yang tidak mengizinkan rentang yang menyertakan nilai tersebut. Pada kasus ini range errornya dilemparkan untuk mencetak atau menampilkan messagenya.

Percobaan 12



Pada percobaan 12 penggunaan proses throw dan catch pada class yang extends exception. Dapat dilihat bahwa terjadi beberapa kali pelemparan hingga pada akhirnya dapat itampilkan apa yang telah di set pada catch yang telah ditentukan.