

NAMA: SITI NURVIATIKA

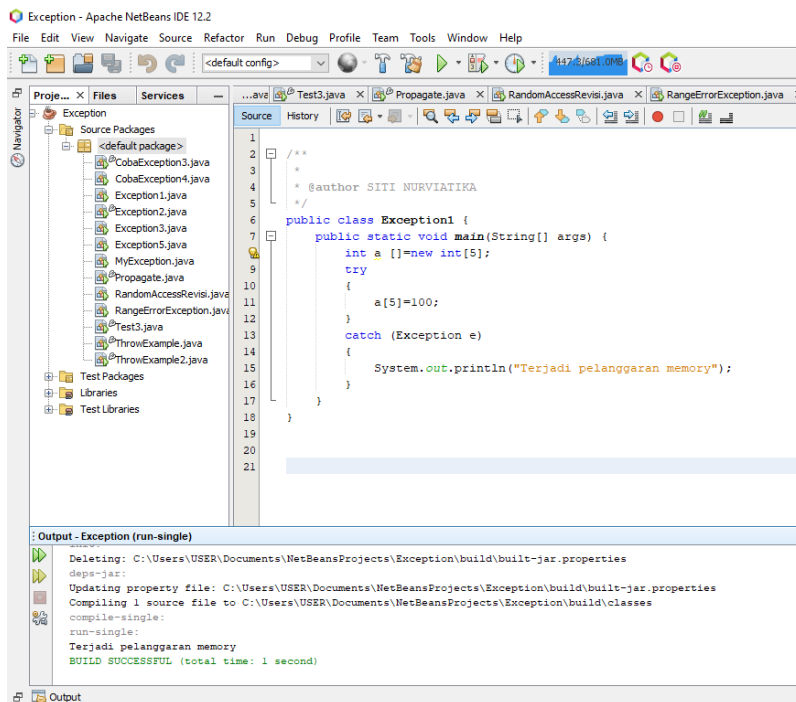
KELAS: TI22A

NIM: 20220040281

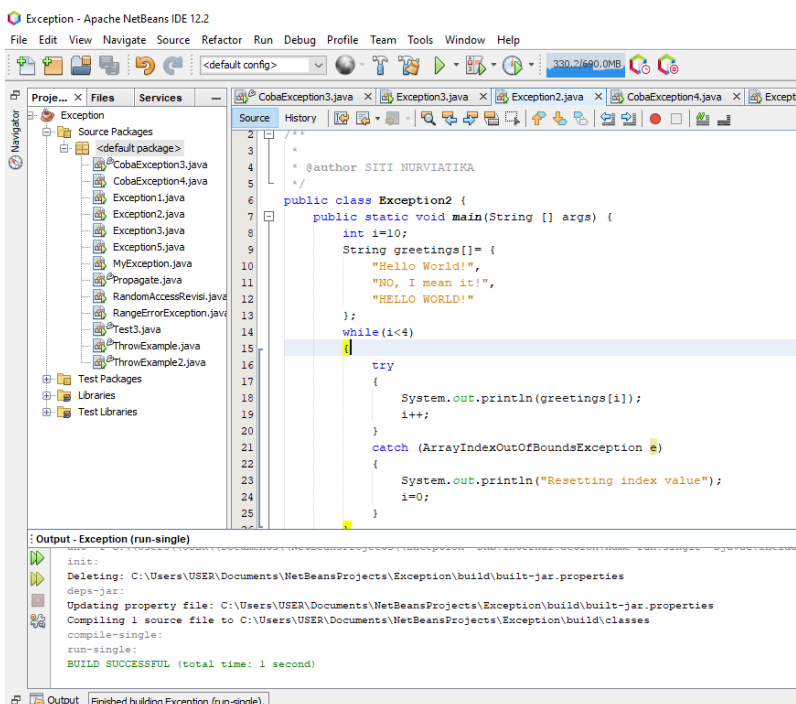
## TUGAS 10

### PBO

#### 1. Percobaan 1



#### 2. Percobaan 2



### 3. Percobaan 3

Exception - Apache NetBeans IDE 12.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config> 474.8/690.0MB

Project: Exception

Source Packages

- <default package>
- CobaException3.java
- CobaException4.java
- Exception1.java
- Exception2.java
- Exception3.java
- Exception5.java
- MyException.java
- Propagate.java
- RandomAccessRevisi.java
- RangeErrorException.java
- Test3.java
- ThrowExample.java
- ThrowExample2.java

Test Packages

Libraries

Test Libraries

```
1  /**
2  *
3  * @author Siti nurviatika
4  */
5
6  public class Exception3 {
7      public static void main(String[] args) {
8          int bil=10;
9          try
10         {
11             System.out.println(bil/0);
12         }
13         catch (Exception e)
14         {
15             System.out.println("Ini menghandle error yang terjadi");
16         }
17     }
18 }
19
20
```

Output - Exception (run-single)

Deleting: C:\Users\USER\Documents\NetBeansProjects\Exception\build\build-jar.properties

deps-jar:

Updating property file: C:\Users\USER\Documents\NetBeansProjects\Exception\build\build-jar.properties

Compiling 1 source file to C:\Users\USER\Documents\NetBeansProjects\Exception\build\classes

compile-single:

run-single:

Ini menghandle error yang terjadi

BUILD SUCCESSFUL (total time: 2 seconds)

Output Finished building Exception (run-single).

Exception - Apache NetBeans IDE 12.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config> 189.5/630.0MB

Project: Exception

Source Packages

- <default package>
- CobaException3.java
- CobaException4.java
- Exception1.java
- Exception2.java
- Exception3.java
- Exception5.java
- MyException.java
- Propagate.java
- RandomAccessRevisi.java
- RangeErrorException.java
- Test3.java
- ThrowExample.java
- ThrowExample2.java

Test Packages

Libraries

Test Libraries

```
1  /**
2  *
3  * @author USER
4  */
5
6  public class CobaException3 {
7      public static void main(String[] args) {
8          int bil=10;
9          try
10         {
11             System.out.println(bil/10);
12         }
13         catch (ArithmeticException e)
14         {
15             System.out.println("Terjadi Aritmatika error");
16         }
17         catch (Exception e)
18         {
19             System.out.println("Ini mengandle error yang terjadi");
20         }
21     }
22 }
23
24
```

Output - Exception (run-single)

Deleting: C:\Users\USER\Documents\NetBeansProjects\Exception\build\build-jar.properties

deps-jar:

Updating property file: C:\Users\USER\Documents\NetBeansProjects\Exception\build\build-jar.properties

Compiling 1 source file to C:\Users\USER\Documents\NetBeansProjects\Exception\build\classes

compile-single:

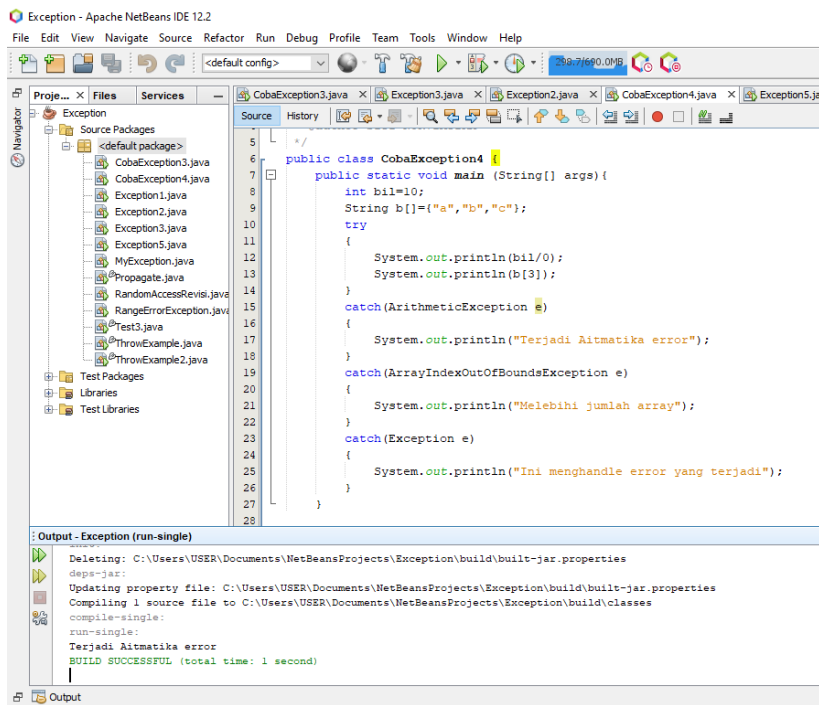
run-single:

1

BUILD SUCCESSFUL (total time: 3 seconds)

Output Finished building Exception (run-single).

#### 4. Percobaan 4



#### 5. Menganalisis blok try dan catch

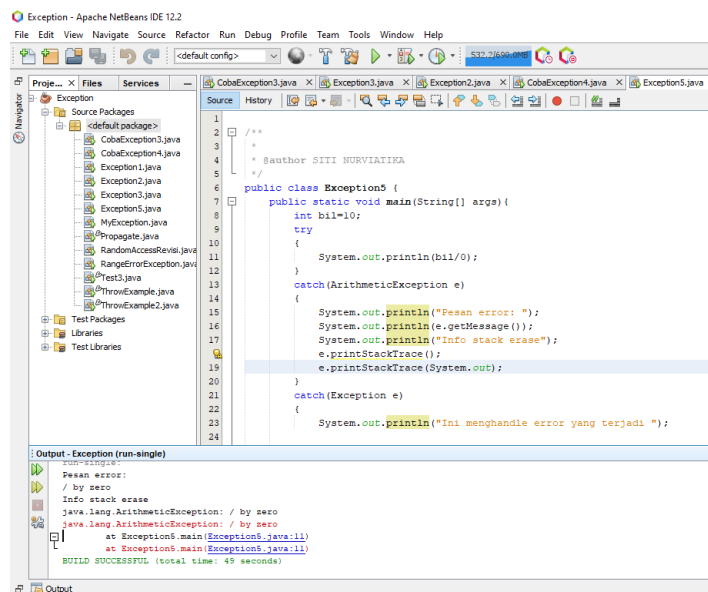
Yang berguna untuk menangani kesalahan yang mungkin akan terjadi saat melakukan pembagian oleh 0.

- **Try**

Dimana 'Try' mencoba melakukan pembagian `'bil/0'` yang merupakan operasi yang tidak valid pada mtk sebab tidak bisa melakukan pembagian oleh 0, yang menyebabkan terjadinya `'ArithmeticException'`.

- **Catch**

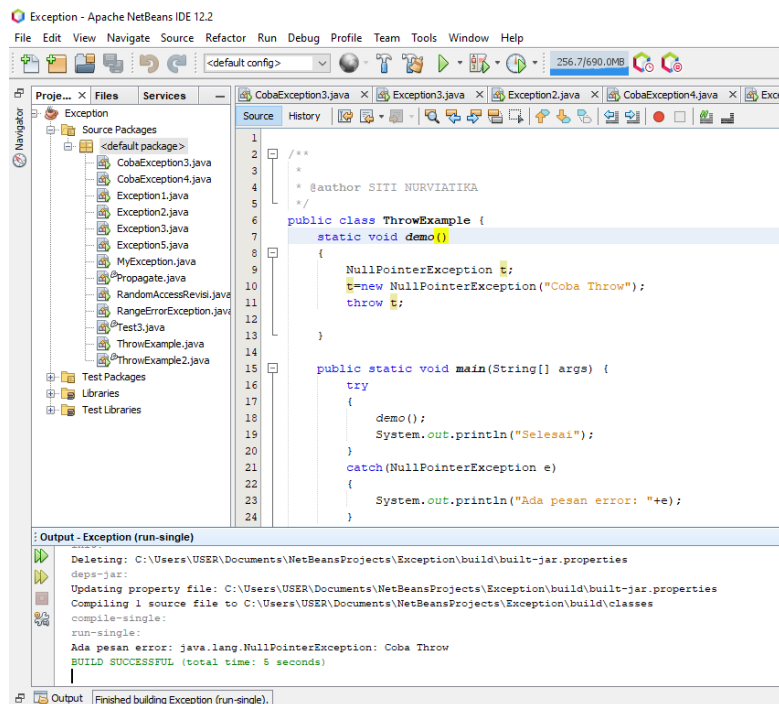
Pada codingan ini menggunakan 2 'Catch' dimana yang pertama untuk mencetak pesan kesalahan yang ada pada object `'ArithmeticException'`. Dan yang kedua untuk menangkap `'Exception'` yang dimana hanya dicetak pesan seperti ini `'Ini menhandle error yang terjadi'`. Ada juga `'e.printStackTrace()'` berguna untuk mencetak informasi stack trace ke dalam konsol. Dan `'e.printStackTrace(System.out)'` untuk mencetak informasi stack trace ke dalam konsol.



## 6. Menganalisis blok try dan catch

Untuk menangani pengecualian yang dilempar oleh method '**demo()**'

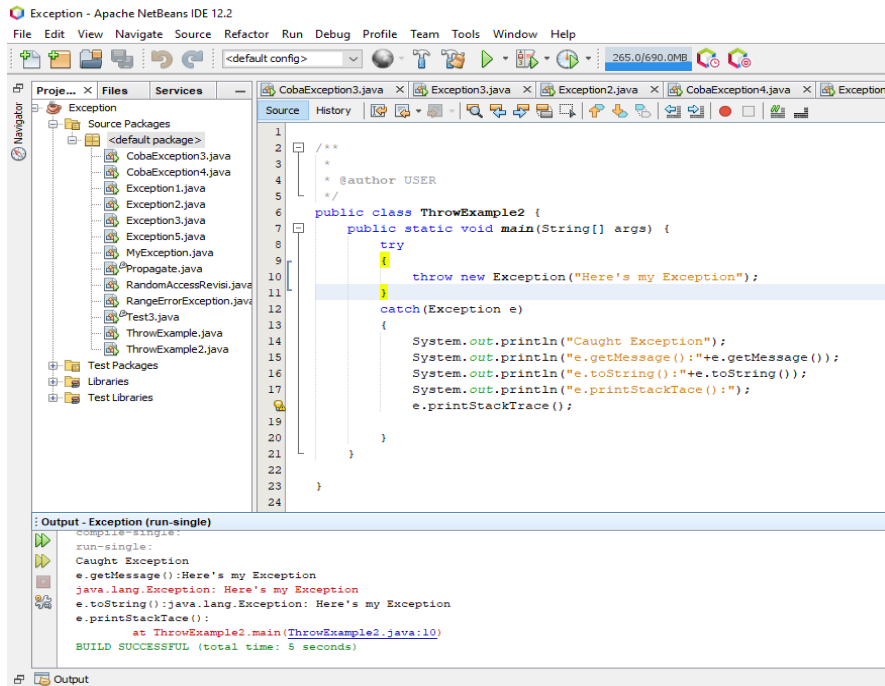
- **Demo()**  
Untuk menganalisis objek '**NullPointerException**' dengan menggunakan pesan '**Coba Throw**', lalu '**demo()**' akan melemparkan pengecualian tersebut menggunakan pernyataan '**throw**'.
- **Try-Catch didalam main()**  
Pemanggilan '**demo()**' dilakukan dalam blok '**try**' jika pengecualian '**NullPointerException**' dilempar oleh '**demo()**' maka '**catch**' akan menangkapnya agar pesan error tidak muncul.



## 7. Menganalisis blok try dan catch

Penggunaan blok ini untuk menangani sebuah pengecualian.

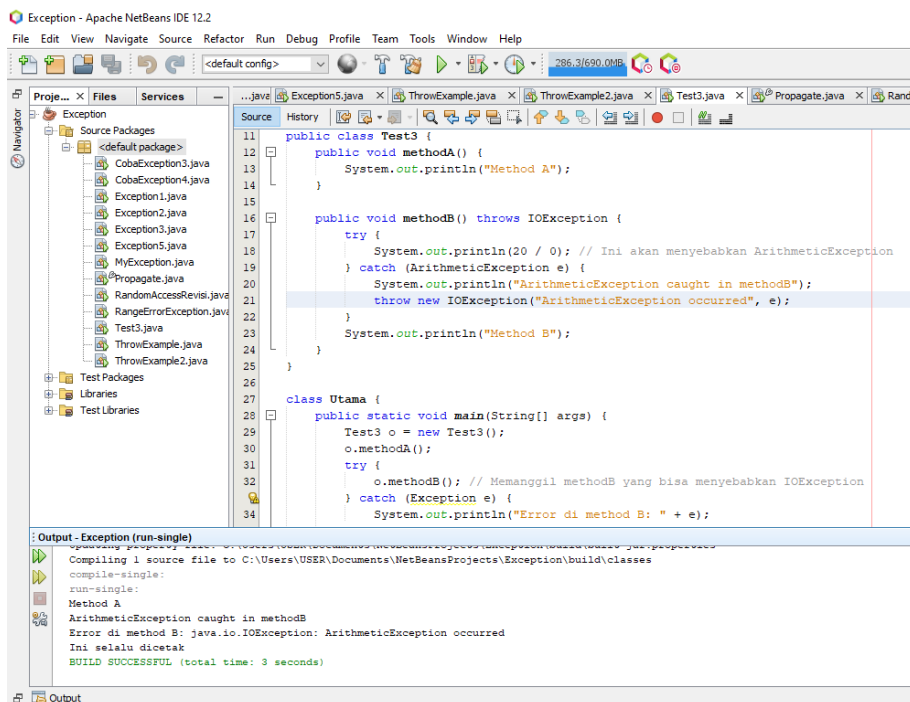
- **Try**  
Ada sebuah pernyataan berupa '**throw new Exception ("Here's my Exception")**' dimana secara eksplisit melemparkan objek '**Exception**', untuk membuat sebuah pengecualian. Dan saat '**throw new Exception ("Here's my Exception")**' dijalankan program ini akan mengeluarkan blok '**try**' dan tidak mengeksekusi pernyataan berikutnya.
- **Catch**  
Pada '**catch (Exception e)**' akan menangkap objek pengecualian yang akan dilempar ke dalam blok '**try**' lalu pengecualian akan ditangkap blok '**catch**' untuk dieksekusi dan mencetak pesan '**Caught Exception**'. Dan kita mencetak informasi tentang pengecualian yang ditangkap menggunakan beberapa metode. Seperti '**e.getMessage()**' mencetak pesan yang ada pada pengecualian, dan '**e.toString()**' mencetak representasi String pengecualian yang ada nama kelas dan pesan, sedangkan '**e.printStackTrace()**' mencetak stack trace pengecualian begitu juga dengan detail lokasi pengecualian terjadi.



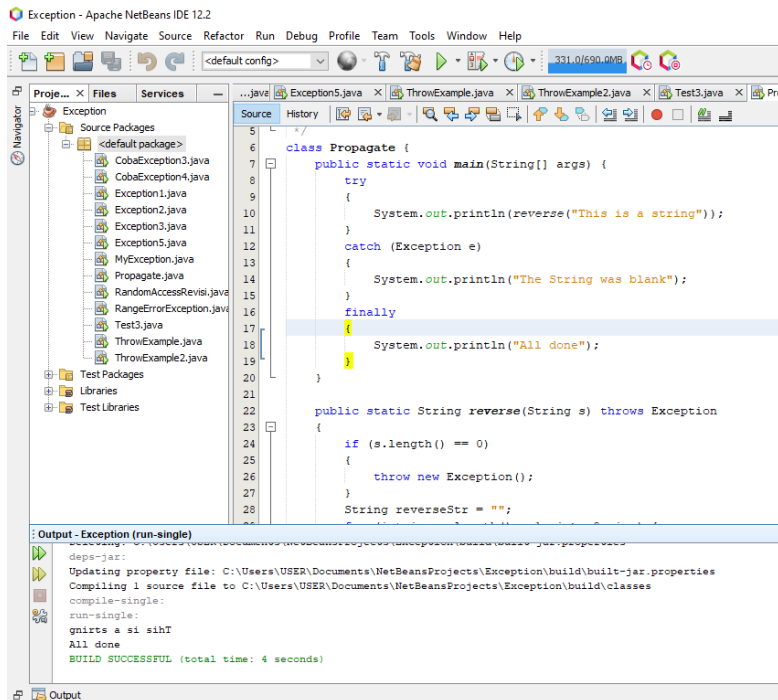
## 8. Analisis penggunaan throws pada pemrograman berikut

**Throw** itu sendiri pada java digunakan untuk memberi tahu pemanggil bahwa sebuah metode bisa melempar pengecualian tertentu.

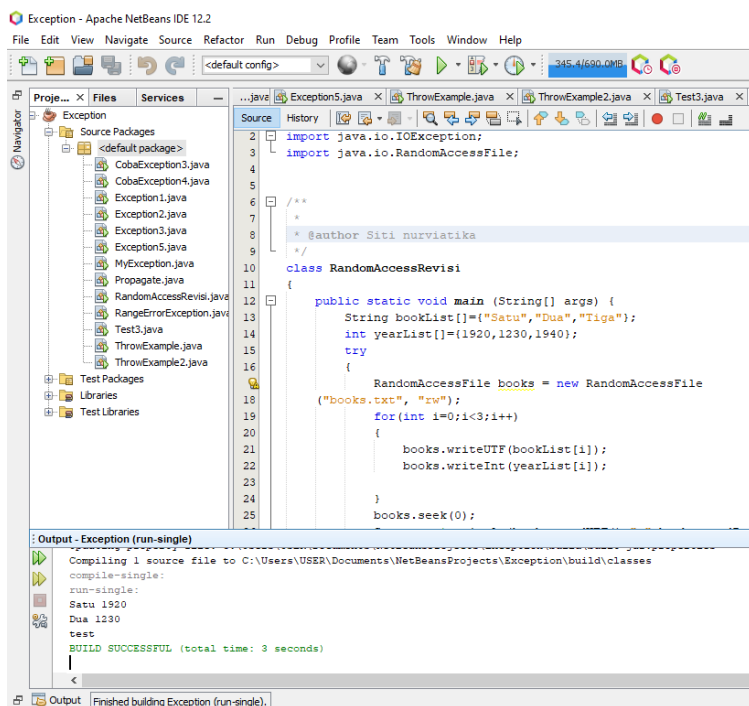
Jadi pada codingan ini **'throws IOException'** pada sebuah deklarasi **'methodB'** itu untuk memberi tahu pemanggil bahwa method tersebut melempar pengecualian tipe **'IOException'**. Jadi pemanggil **'methodB'** harus menangani pengecualian tersebut seperti pengecualian dalam blok **'try-Catch'** pada metode **'main'**.



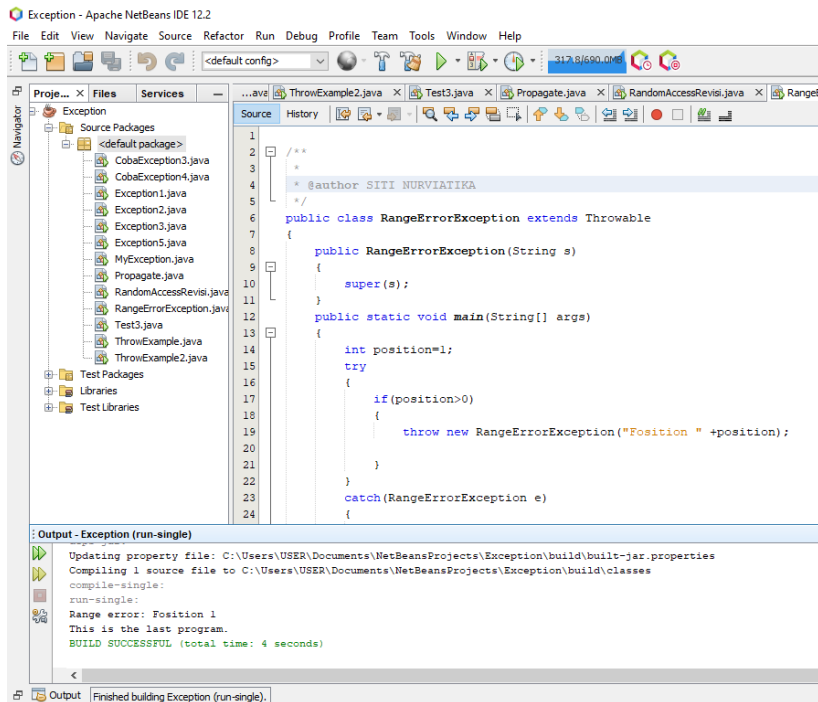
## 9. Percobaan 9 membalik String (reverse)



## 10. Percobaan 10



## 11. Percobaan 11



## 12. Percobaan 12

