**Percobaan 1**



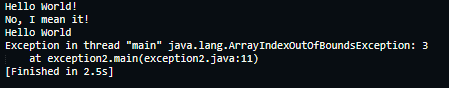
Error terjadi karena array bersize 5 dari 0-4, namun program mencoba menset value ke indeks ke 5.

**Pembetulan**

****

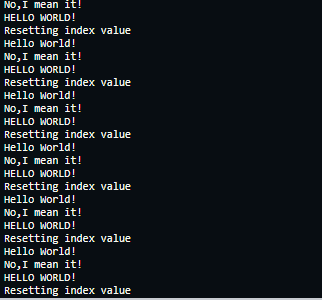
Karena terdapat blok try{} dan catch{} sehingga error diatas dapat kita handle sesuai keinginan kita.

**Percobaan 2**



Masalah sama seperti di percobaan 1 yaitu indeks array hanya 0 - 2 tapi mencoba mengakses indeks ke-3.

**Pembetulan**



Terjadi infinite loop karena saat terjadi exception interasi (int i) akan direset ke 0 sehingga while akan terus berada dalam kondisi TRUE.

**Percobaan 3**



Error terjadi karena bilangan tidak bisa dibagi oleh 0.

**Pembetulan**



Error tersebut dihandle dengan try catch.

**Percobaan 4**



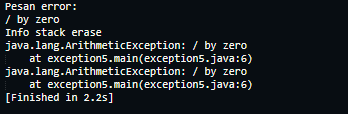
Exception yang dihandle adalah ArrayIndexOutOfBoundsException.

**Pembetulan**



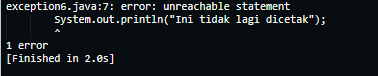
Exception yang dihandle adalah ArithmeticException, berarti try akan menthrow exception yang terjadi pertama yang kemudian akan dihandle oleh blok catch{} sesuai dengan parameter exceptionnya(yang lebih spesifik).

**Percobaan 5**



Pesan error adalah saat kita memanggil method **getMessage()** dan Info stack erase adalah saat kita memanggil method **printStackTrace().**

**Percobaan 6**



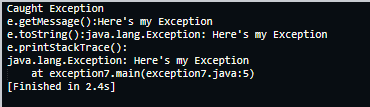
Terjadi error unreachable karena method demo() sudah melempar sebuah throw.

*Coba membenarkan dengan menghilangkan perintah yang unreachable.*



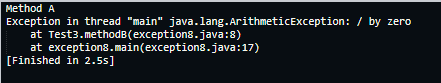
Method demo() menthrow sebuah NullPointerException yang tersimpan di variable t dengan parameter “Coba Throw” yang akan menjadi message dari exception tersebut. Exception tersebut dihandle dengan blok catch{} di main method.

**Percobaan 7**



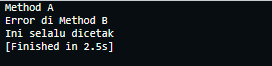
Blok try{} melempar sebuah Exception yang diberi parameter “Here’s my Exception” (menjadi message dari Exception tersebut) kemudian ditangkap dan dihandle oleh blok catch{} dengan mencoba method-method yang ada pada Exception.

**Percobaan 8**



Terjadi Exception pada method B karena sebuah angka dibagi dengan 0 dan tidak dihandle oleh si pemanggil.

**Pembetulan**



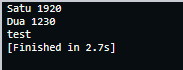
Exception pada method B sudah dihandle dengan blok catch{} dengan print(“Error di Method B”) kemudian terdapat blok finally{} yang akan selalu dieksekusi yang menampilkan “Ini selalu dicetak”.

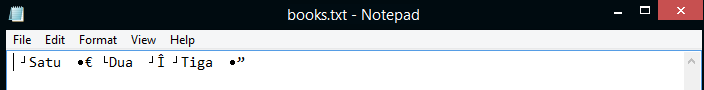
**Percobaan 9**



Tampil “The String was blank” karena method reverse() melempar Exception jika string lengthnya = 0. Exception tersebut dihandle oleh blok catch{}.

**Percobaan 10**





Program berhasil dan tidak ada IOException yang terjadi.



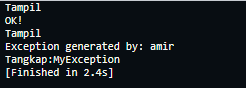
**IOException terjadi** saat parameter kedua dari RandomAccessFile adalah “r”(read) karena hal tersebut akan menthrow sebuah IOException apabila kita mencoba meng-write-nya.

**Percobaan 11**



Program diatas menunjukkan proses throw and catch pada class yang extends Throwable.

**Percobaan 12**



Program diatas menunjukkan proses throw and catch pada class yang extends Exception.

**Kesimpulan**

Kelas Exception merupakan kelas turunan dari kelas Throwable di package Java.Lang. Exception merupakan kesalahan yang dapat dihandle oleh developer dengan perintah seperti try dan catch. Hal ini sangat berguna karena apabila terjadi masalah dan sudah dihandle, program tidak akan berhenti alias tetap berjalan.