Jobsheet 5

Nama : Aulya Dina Rifqiah

Kelas : XI RPL 5

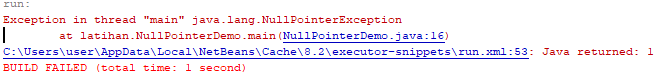
No : 07

**Latihan 1 *(NullPointerException)***

String name = “isi nama masing”;

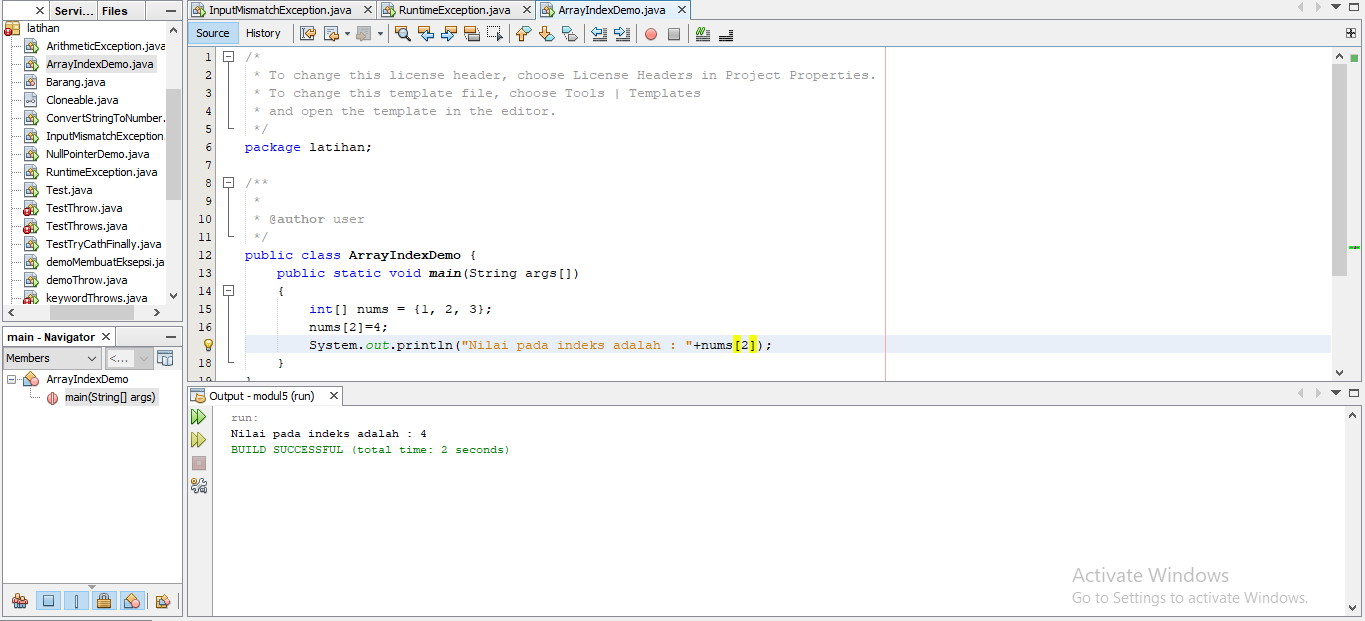


String name = null;

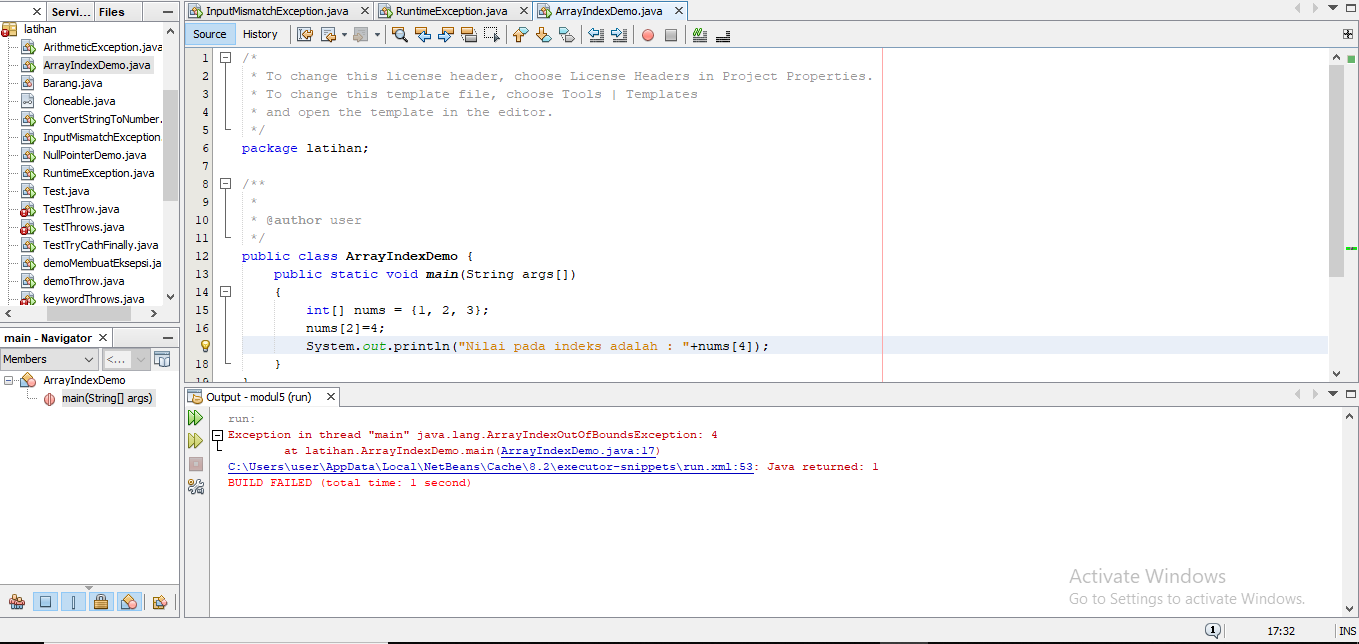


**Latihan 2 *(ArrayIndexOutOfBoundsException)***

System.out.println("Nilai pada indeks adalah : "+nums[**2**]);

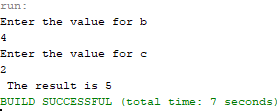


System.out.println("Nilai pada indeks adalah : "+nums[**4**]);

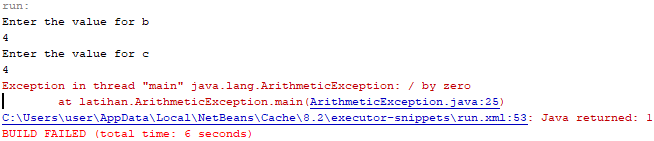


**Latihan 3 *(ArithmeticException)***

Setelah dijalankan inputkan nilai b = 4 dan c = 2; Apa yang terjadi..?

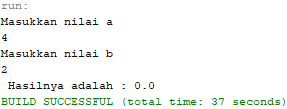


Kemudian jalankan kembali dan inputkan nilai b = 4 dan c = 4; Apa yang terjadi? Jelaskan!.

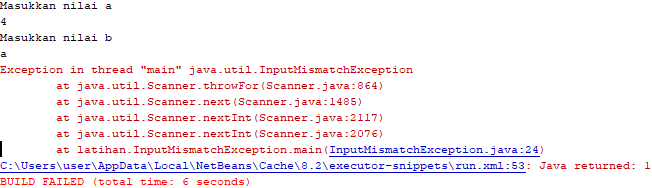


**Latihan 4 *(InputMismatchException)***

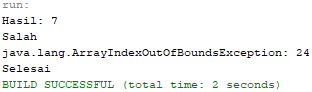
Setelah dijalankan inputkan nilai b = 4 dan c = 2; Apa yang terjadi..?



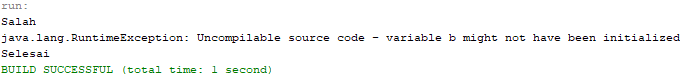
Kemudian jalankan kembali dan inputkan nilai b = 4 dan c = a; Apa yang terjadi? Jelaskan!



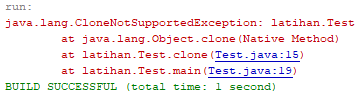
**Latihan 5 *(RuntimeException)***



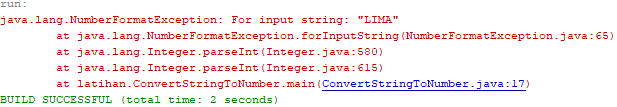
Kemudian jalankan kembali dan ganti int b = 5;; Apa yang terjadi?



**Latihan 6 *(CloneNotSupportedException)***



**Latihan 7 *(NumberFormatException)***



**Tugas Praktikum**

1. Jelaskan perbedaan penggunaan try catch finally dan throw throws!

Perbedaan penggunaan try catch finally dan throw throws:

Try catch finally, blok try – catch dan finally yang memiliki proses yang lebih lengkap, karena pada finally kita dapat mendefinisikan kode program yang selalu dieksekusi, baik ada exception yang terjadi maupun bila tidak terjadi exception sama sekali.

Throw, Eksekusi program akan dihentikan segera setelah kalimat throw. Java akan melakukan inspeksi blok try terdekat untuk menemukan catch yang cocok dengan dengan tipe exception yang dilempar. Jika Java menemukan, maka kendali program ditransfer ke kalimat itu. Jika tidak ditemukan, maka Java akan melakukan penelusuran ke blok berikutnya dan bila tetap tidak ditemukan, maka penanganan exception secara default akan menghentikan programnya.

Throws, Jika sebuah method dapat menyebabkan sebuah exception namun tidak menangkapnya, maka digunakan keyword throws. Sebuah method perlu untuk menangkap ataupun mendaftar seluruh exceptions yang mungkin terjadi, namun hal itu dapat menghilangkan tipe Error, RuntimeException, ataupun subclass-nya.

1. Jelaskan dengan bahasa dan pemahanmu sendiri mengenai jenis-jenis exception yang ada!

Try: Pemunculan program yang ingin di proses

Catch : untuk memproses program dari kesalahan

Finally : untuk pengeluaran meskipun itu ada kesalahan

Throw : *Keyword****throw*** digunakan untuk melempar suatu eksepsi dalam *program*

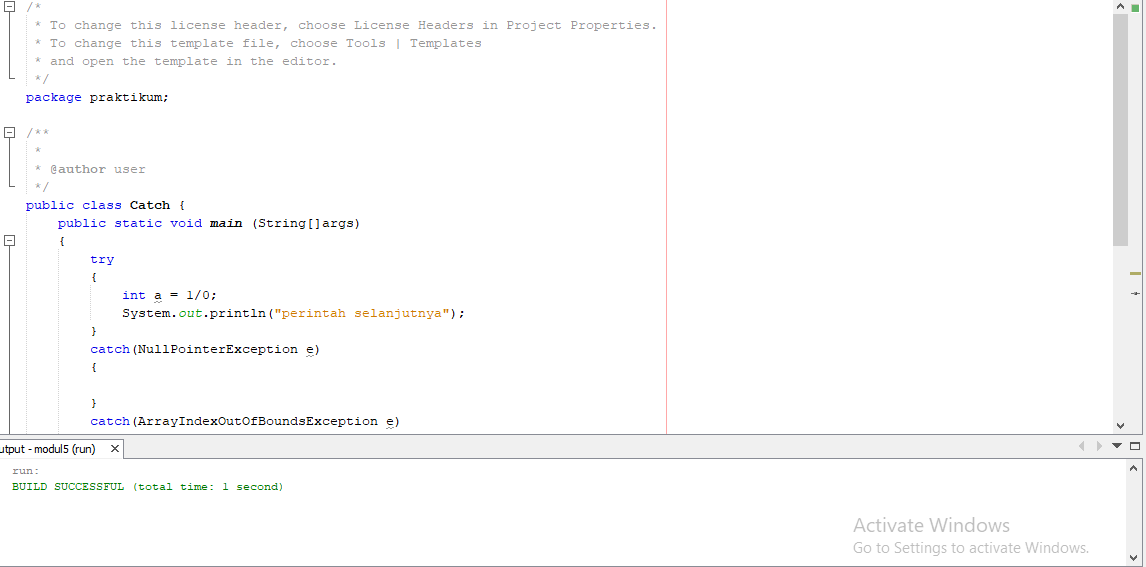
Throws : *Keyword****throws*** digunakan pada waktu mendeklarasikan suatu *method* untuk memberitahu bahwa *method* yang bersangkutan dapat melempar eksepsi dengan tipe yang dideklarasikan oleh *keyword****throws*** tersebut.

1. Buatlah dua program exception handling dengan menggunakan try catch finally dan throw throws (2 jenis exception)!

Try



Catch



Final



Throw



Throws

