**LAPORAN PRAKTIKUM**

**DASAR-DASAR PEMROGRAMAN PERTEMUAN KE-6**

Penanganan Kesalahan dan Input

****

Disusun Oleh:

Muhammad Fitroh Amrilla

(11190910000022)

Program Studi Teknik Informatika

Fakultas Sains dan Teknologi

2019/2020

1. **Pembahasan**
2. Mengetahui arti eksepsi(kesalahan) dalam Java
3. Mengetahui jenis- jenis kesalahan dalam Java
4. Mengetahui dasar- dasar penanganan kesalahan dalam Java
5. Mengetahui tipe- tipe kesalahan dalam Java
6. Mengetahui jenis-jenis input dalam Java
7. **Dasar Teori**

Eksepsi dalam Java adalah keadaan tidak normal yang muncul pada suatu bagian pogram yang dijalankan. Penangan eksepsi di java membawa pengelolaan kesalahan program saat dijalankan ke dalam orientasi- objek. Eksepsi java adlah objek yang menjelaskan suatu keadaan eksepsi yang muncul pada suatu bagian program.

Ada 2 jenis kesalahan, yaitu kesalahan pada saat kompilasi program dan kesalahan ketika proram dijalankan. Akan tetapi, bila kesalahan terjadi saat program berjalan, maka program akan menjadi error. Oleh karena itu, perlu adanya mekanisme penanganan kesalahan.

Penanganan kesalahan dalam java daiatur olleh 5 kata kunci : try, catch, throw, thorws, dan finally. Pada dasarnya *try* digunakan untuk mengeksekusi suatu bagian program, dan jika muncul kesalahan, sistem akan melakukan *throw* suatu kesalahan yang dapat di *catch* berdasarkan tipe kesalahannya, atau bisa diberikan *finally* dengan penanganan *default*.

Dipuncak hirarki eksepsi terdapat suatu class yag disebut *Throwable*. Calss ini digunakan untuk merepresentasikan semua keadaan eksepsi. Class Thorwable terbagi menjadi 2 bagian. 1. *Class Exception* digunakan untuk keadaan eksepsi yang harus ditangkap oleh program yang dibuat. Sedangkan yang lain diharapkan dapat menangkap class yang kita subclass kan untuk menghasilkan keadaaan eksepsi.

2. *Class Error* , yang mendefinisikan keadaan yantg tidak dihaarapkan untuk ditangkap dalam lingkungan normal.

**Try**  adalah kata kunci yang digunakan untuk mencoba menajalankan block program kemudian mengenai dimana munculnya kesalahan yang ingin diproses. Keyword try harus dipasangkan dengan keyword catch atau finally.

**Catch**  adalah kata kunci yang berfungsi menangkap kesalahan atau bug yang terjadi dalam blok try. Keyword catch dapat diletakan berulang-ulang sesuai kebutuhan.

**Finally**  adalah keyword yang menunjukan bahwa blok program tersebut akan selalu dieksekusi meskipun ada kesalahan yang muncul atau pun tidak ada. Berbeda dengan keyword catch, keyword finally hanya dapat diletakan satu kali setelah keywrd try.

**Throw** digunakan untuk melemparkan suatu eksepsi atau bug yang dibuat secara manual.

**Throws**  adalah keyword yang digunakan dalam suatu method yang ungkin menghasilkan suatu kesalahan sehingga perlu ditangkap errornya.

Ada tiga input yang bisa kita pakai dalam java yaitu :

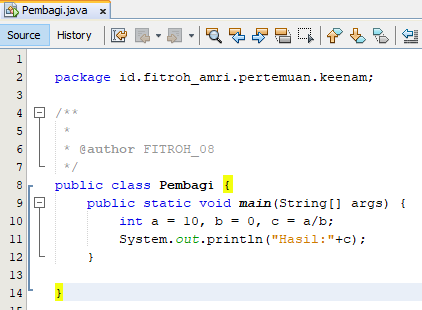
1. **Scanner**. Untuk memnggunakan Scanner ini , diperlukan untuk mmebuat objek dari class Scanner, class tersebut berada di dalam package *java.util.*

2. **Buffered Reader** adalah abstrak yang menangani baca tulis ke suatu media. Kelas ini membutuhkan kelas lain sebagai pekerjanya, yaitu InputStreamReader. Dan InputStreamReader membutukan media tempat bacatulis dilakukan, yaitu System.in. Buffered Reder berada di dalam package *java.io*. Fungsi Buffered Reader ini hampira sama dengan Scanner hanya saja terdapat perbedaan dalam penulisan syntax nya.

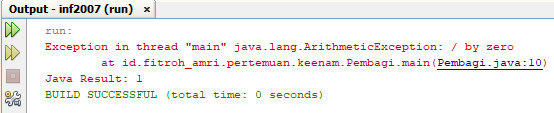
3. **JoptionPane** merupakan package dari javax.swing, digunakan untuk input dan output berbasis GUI swing, dan tampilannya memudahkan pengguna karena berupa dialog box standar yang memfasilitasi kepada user untuk mengkonfirmasi sesuatu.

1. **Latihan**
2. Membuat program Pembagi.

Source Code :

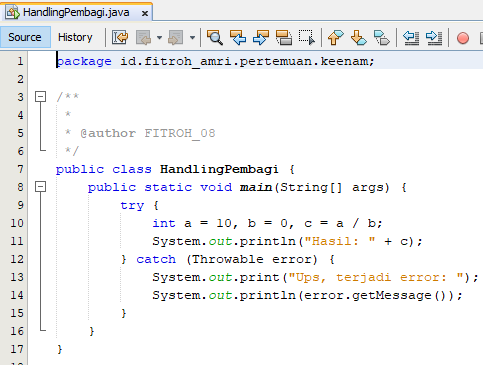


Output : output memiliki kesalahan yaitu pembagi sama dengan nol (0)

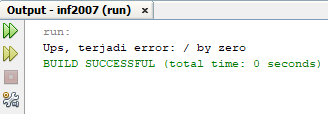


1. Membuat program yang menangani error no 1 dengan try dan catch.

Source Code :

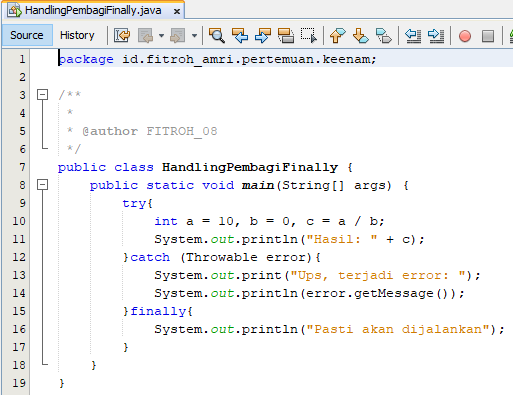


Output :

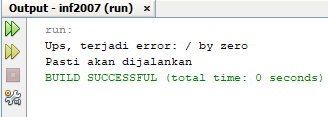


1. Membuat program yang menangani error no 1 dengan try, catch dan finally .

Source Code :

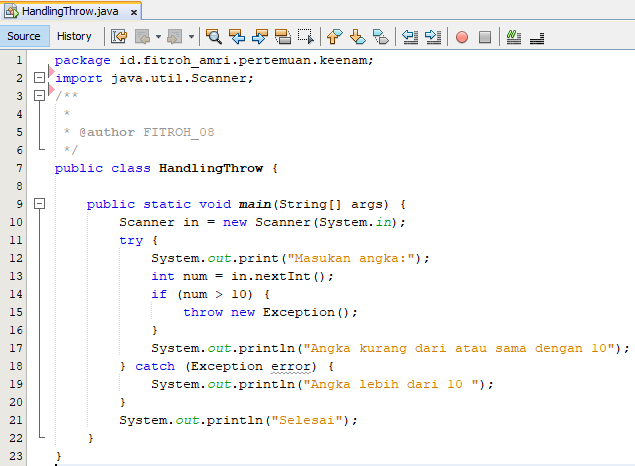


Output :



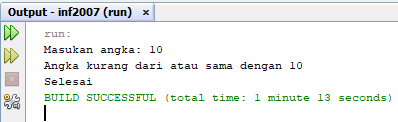
1. Membuat program yang menangani kesalahan dengan throw

Source Code :

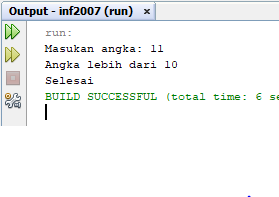


Output :

1.

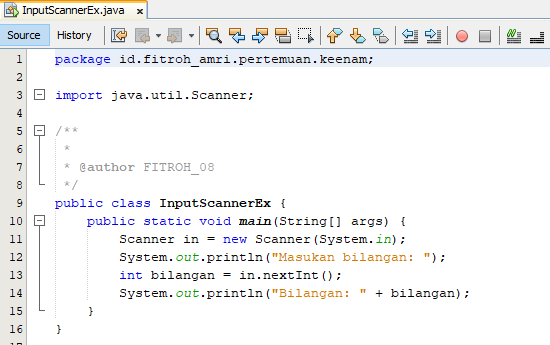


2.

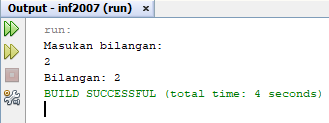


1. Menginput nilai menggunakan scanner.

Source Code :

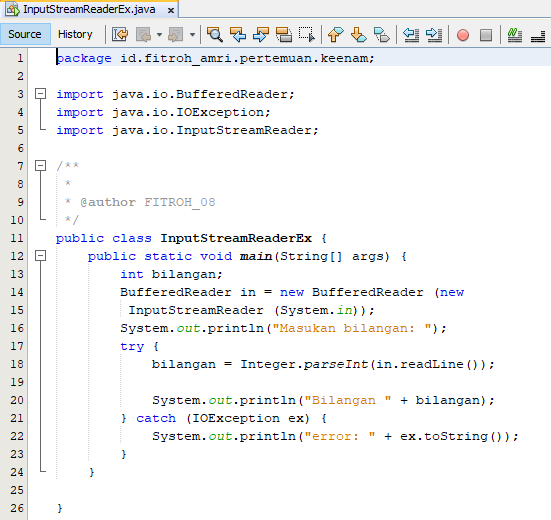


Output :

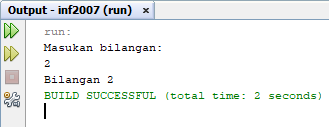


1. Menginput nilai menggunakan stream reader .

Source Code :

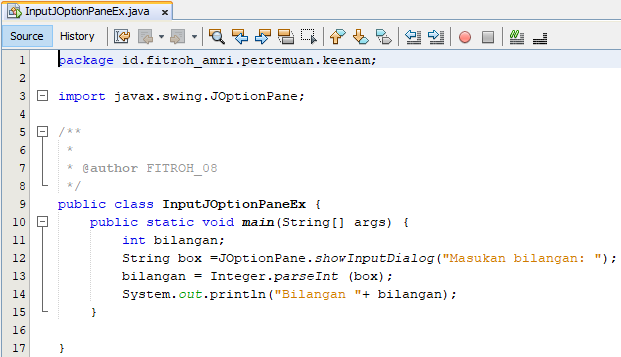


Output :

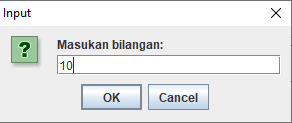


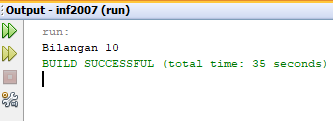
1. Menginput nilai menggunakan JOption.

Source Code :



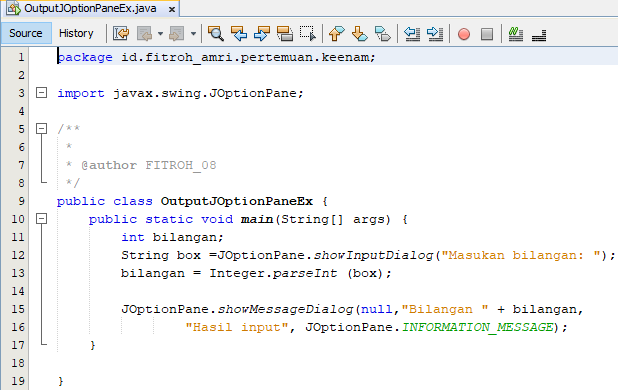
Output :





1. Menginput nilai menggunakan JOption.

Source Code :



Output :

