1. **DEFINISI MASALAH**
2. Buatlah sebuah class turunan dari Exception baik di Java maupun C++ (pilih salah satu bahasa sesuai apa yang digunakan di praktikum) yang digunakan sebagai tipe exception yang dapat dilemparkan ketika menghitung n faktorial dimana n < 0. Buat method untuk menghitung faktorial terlebih dahulu!
3. Lakukan reproduksi exception dan handlingnya dalam beberapa kasus dibawah (pilih minimal 2)
   1. I/O Exception
   2. Stack Overflow Exception
   3. Array index out of bound exception
4. **SOURCE CODE**
   1. **Soal Pertama**

|  |  |
| --- | --- |
|  | Main.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | import java.util.Scanner;  /\*\*  \* Created by Zain Fikri H on 23/05/2017.  \*/  public class Main extends Exception {  public static void main(String[] args) {  Scanner in = new Scanner(System.in);  System.out.println("Sistem Faktorial");  System.out.print("Masukkan Bilangan : ");  int n = in.nextInt();  try {  System.out.println(faktorial(n));  }catch (Exception e){  System.out.println(e.getMessage());  }  }  public static int faktorial(int n) throws Exception{  if(n<0){  throw new Exception("n kurang dari 0");  }if(n==0||n==1){  return 1;  }else {  return n\*faktorial(n-1);  }  }  } |

* 1. **Soal Kedua**

|  |  |
| --- | --- |
|  | Main2.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | */\*\*  \* Created by Zain Fikri H on 25/05/2017.  \*/* public class Main2 {  public static void main(String[] args) {  *//IndexOutOfBoundsException* String string = new String("Hai, Nama Saya Zain Fikri Hanastyono");  System.*out*.println("Contoh Index Out Of Bounds");  try{  System.*out*.println(*charAt*(100, string));  }catch (IndexOutOfBoundsException e){  System.*out*.println("Error : " + e.getMessage());  }  *//stackOverflow* System.*out*.println("\nContoh Handling Stack Overflow");  *cekStack*();  }   *//method stackOverflow* public static void stack(){  *stack*();  }   *//handling stackOverflowError* public static void cekStack() throws StackOverflowError{  try {  *stack*();  }catch (StackOverflowError e){  System.*out*.println("Error : Stack Overflow");  }  }   *//handling indexOfOutBoundsException* public static char charAt(int index, String string) throws IndexOutOfBoundsException {  if(index>string.length()){  throw new IndexOutOfBoundsException("Maaf Index Tidak Ditemukan");  }return string.charAt(index);  } } |

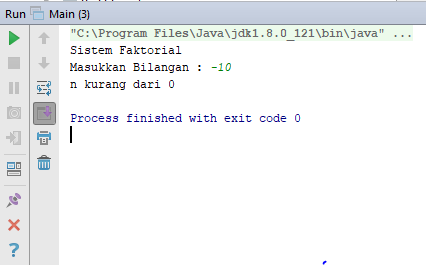
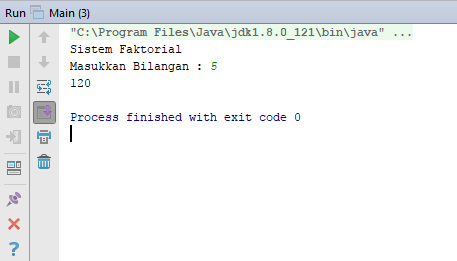
1. **PEMBAHASAN**
   1. **Soal Pertama**

|  |  |
| --- | --- |
|  | Induk.java |
| 1  6  7  8  9  10  11  12  13  14  15  18  19  20  21  22  23  24 | Import java util scanner  Deklarasi class Main sebagai turunan dari Exception  Deklarasi main method  Inisialisasi Scanner  Print “Sistem Faktorial  Print “Masukkan Bilangan : “  Deklarasi variabel n dan inpuran user  Statement try  Print nilai dari factorial(n)  Statement catch dengan parameter Exception e  Print nilai dari e.gerMessage()  Deklarasi method factorial dengan parameter integer n sebagai Exception  Seleksi kondisi if dengan kondisi n<0  Statement pelemparan exceprtion “n kurang dari 0”  Seleksi kondisi if dengan kondisi n sama dengan 0 atau n sama dengan 1  Statement return 1  Seleksi kondisi else  Statement return n dikali nilai factorial(n-1) |

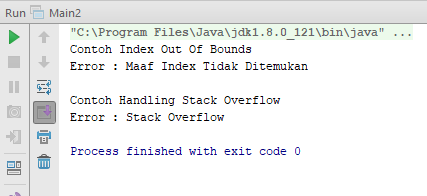
* 1. **Soal Kedua**

|  |  |
| --- | --- |
|  | Induk.java |
| 4  5  7  9  10  11  12  13  17  19  23  24  28  30  31  32  33  38  40  41  43 | Deklarasi class Main2  Deklarasi main method  Deklarasi dan inisialisasi variabel string bertipe String  Print “Contoh Index Out Of Bounds”  Statement try  Print nilai dari method charAt berparameter  Statement catch dengan parameter IndexOfOutBoundsException  Print “Error : “ dan nilai dari method getMessage  Print “Contoh Handling Stack Overflow”  Statement menjalankan method cekStack  Deklarasi method stack  Statement menjalankan method stack  Deklarasi method cekStack sebagai turunan dari StackOverflowError  Statement try  Menjalankan method stack  Statement catch dengan parameter StackOverflowError  Print “Error : Stack Overflow”  Deklarasi method charAt berparameter sebagai turunan dari IndexOutOfBoundsException  Seleksi kondisi if dengan kondisi parameter index lebih dari panjang parameter string  Statement throw baru dengan nilai “Maaf Index Tidak Ditemukan”  Statement return dengan nilai string.charAt(index) |

1. **SCREENSHOT PROGRAM**
   1. **Soal Pertama**

****

* 1. **Soal Kedua**

****

1. **PRAKTIKUM**
2. **Statis Method**
3. Jalankan kode program dan benahi jika ada kesalahan!

*Jawaban:*

Terdapat kesalahan dalam penulisan tanda kutip pada perintah print. Terdapat kesalahan pada variabel string yy karena belum diinisialisasi. Dan ada kesalahan pada penuslian symbol pengurangan.

1. Jelaskan secara lengkap bagaimana pembenahan kesalahan yang telah dilakukan di langkah pertama!

*Jawaban:*

Pertama saya membetulkan tanda kutip yang sesuai untuk string. Kemudian saya membetulkan dengan menginisialisasi pada variabel string yy. Dan saya membetulkan dengan mengganti symbol pengurangan yang sesuai.

1. (Java) jika pada method division(int, int) statement throws Exception dihapus, apa yang terjadi? Jelaskan secara lengkap!

*Jawaban:*

Tidak terjadi kesalahan dan output yang dihasilakan sama dengan dengan output sebelum throws Exception dihapus.

1. Jelaskan secara lengkap menurut pendapat Anda mengenai null pointer exception, bagaimana bisa terjadi, bagaimana mereproduksi exception tersebut, dan penanggulannya!

*Jawaban:*

Null pointer exception terjadi karena terdapat variabel atau method yang belum diinisialisasi sehingga saat program dijalankan akan terjadi error. Hal ini ditangani dengan cara memberikan keterangan bahwa ada variabel atau method yang null dengan cara pengecekan apakah suatu variabel atau method sama dengan null, apabila iya maka akan mencetak null pointer exception.

1. **KESIMPULAN**

* Exception merupakan kondisi atau event yang terjadi ketika runtime yang mengganggu alur instruksi pada kode program yang sebenarnya. Exception biasa terjadi jika ada runtime error yang disebabkan oleh: null pointer exception, I/O error, index out of bounds, division by zero dan sebagainya. Ketika exception tidak dihandle dengan baik maka program akan seketika crash
* Exception handling digunakan saat terdapat suatu program dimana yang sudah diprediksi akan terjadi error, namun error tersebut dapat dihandle dengan cara menambahkan suatu handling yang sesuai dengan error yang akan terjadi. Jika suatu program sudah diberi exception handling maka jika ada error yang sesuai dengan handling secara otomatis akan memberikan output berupa nilai dari exception handlingnya dan program akan tetap berjalan.