

Nama : Hilda Deliana A
Nim : 20220040112
Kelas : TI22H

“EXCEPTION HANDLING”

Percobaan 1

Error yang terjadi : “ArrayIndexOutOfBoundsException” karena mencoba mengakses indeks yang tidak ada dalam array.

```
1- public class Exception {
2-     public static void main(String[] args) {
3-         int a[]=new int[5];
4-         try {
5-             a[5]=100;
6-         } catch(Exception e) {
7-             System.out.println("Terjadi pelanggaran memory");
8-         }
9-     }
10 }
11 |
```

Penjelasan:
Menambahkan blok try-catch untuk menangkap ArrayIndexOutOfBoundsException dan menampilkan pesan kesalahan yang lebih jelas.

Percobaan 2

Error yang Terjadi: ‘ArrayIndexOutOfBoundsException’ karena i mencapai nilai 4 yang melebihi batas array.

```
1- public class Exception2 {
2-     public static void main(String[] args) {
3-         int i=0;
4-         String greetings[]={
5-             "Hello World!",
6-             "No,I mean it!",
7-             "HELLO WORLD!"
8-         };
9-         while(i<4) {
10-             try {
11-                 System.out.println(greetings[i]);
12-                 i++;
13-             } catch(ArrayIndexOutOfBoundsException e) {
14-                 System.out.println("Resetting index value");
15-                 i=0;
16-             }
17-         }
18-     }
19 }
```

Penjelasan:
Menambahkan blok try-catch untuk menangkap ArrayIndexOutOfBoundsException dan mengatur ulang indeks i jika terjadi kesalahan.

Percobaan 3

Error yang Terjadi: ‘ArithmeticException’ karena pembagian dengan nol.

```
1- public class Exception3 {
2-     public static void main(String[] args) {
3-         int bil=10;
4-         try {
5-             System.out.println(bil/0);
6-         } catch(Exception e) {
7-             System.out.println("Ini menghandle error yang terjadi");
8-         }
9-     }
10 }
11 |
```

Penjelasan:
Menambahkan blok try-catch untuk menangkap ArithmeticException dan menampilkan pesan kesalahan.

percobaan 4

```
1- public class Exception4 {
2-     public static void main(String[] args) {
3-         int bil=10;
4-         String b[]={"a","b","c"};
5-         try {
6-             System.out.println(bil/0);
7-             System.out.println(b[3]);
8-         } catch(ArithmeticException e) {
9-             System.out.println("Terjadi Aritmatika error");
10-        } catch(ArrayIndexOutOfBoundsException e) {
11-            System.out.println("Melebihi jumlah array");
12-        } catch(Exception e) {
13-            System.out.println("Ini menghandle error yang terjadi");
14-        }
15-    }
16- }
17
```

Penjelasan:

Menukar urutan operasi dalam blok try sehingga ArithmeticException terjadi lebih dulu.

Percobaan 5

```
1- public class Exception5 {
2-     public static void main(String[] args) {
3-         int bil=10;
4-         try {
5-             System.out.println(bil/0);
6-         } catch(ArithmeticException e) {
7-             System.out.println("Pesan error: ");
8-             System.out.println(e.getMessage());
9-             System.out.println("Info stack erase");
10-            e.printStackTrace();
11-            e.printStackTrace(System.out);
12-        } catch(Exception e) {
13-            System.out.println("Ini menghandle error yang terjadi");
14-        }
15-    }
16- }
17
```

Analisa:

Menangkap ArithmeticException, menampilkan pesan error, informasi stack trace, dan menulis stack trace ke output standar. Exception umum menangani kesalahan lain yang mungkin terjadi.

Percobaan 6

```
1- public class ThrowExample {
2-     static void demo() {
3-         NullPointerException t;
4-         t = new NullPointerException("Coba Throw");
5-         throw t;
6-         // Baris ini tidak lagi dikerjakan;
7-         System.out.println("Ini tidak lagi dicetak");
8-     }
9-
10-    public static void main(String[] args) {
11-        try {
12-            demo();
13-            System.out.println("Selesai");
14-        } catch(NullPointerException e) {
15-            System.out.println("Ada pesan error: " + e);
16-        }
17-    }
18- }
```

Analisa:

NullPointerException dilempar manual dengan pesan "Coba Throw". Baris setelah throw tidak akan dijalankan. catch menangkap dan menampilkan pesan error.

Percobaan 7

```
3-      try {
4-          throw new Exception("Here's my Exception");
5-      } catch(Exception e) {
6-          System.out.println("Caught Exception");
7-          System.out.println("e.getMessage():" + e.getMessage());
8-          System.out.println("e.toString():" + e.toString());
9-          System.out.println("e.printStackTrace():" + e.printStackTrace());
10-         e.printStackTrace();
11-     }
12- }
```

Analisa:

Melempar Exception dengan pesan khusus, kemudian menangkapnya dan menampilkan pesan error, representasi string dari exception, dan stack trace.

Percobaan 8

Error yang Terjadi: 'ArithmeticException' dalam methodB tidak ditangani.

```
1- class Exception0 {
2-     public static void main(String[] args) {
3-         Test3 o = new Test3();
4-         o.methodA();
5-         try {
6-             o.methodB();
7-         } catch(Exception e) {
8-             System.out.println("Error di Method B");
9-         } finally {
10-             System.out.println("Ini selalu dicetak");
11-         }
12-     }
13- }
14- }
```

Penjelasan:

Menangani exception dalam main menggunakan blok try-catch dan menambahkan blok finally yang selalu dijalankan.

Percobaan 9

```
1- class Propagate {
2-     public static void main(String[] args) {
3-         try {
4-             System.out.println(reverse("This is a string"));
5-         } catch(Exception e) {
6-             System.out.println("The String was blank");
7-         } finally {
8-             System.out.println("All done");
9-         }
10-     }
11-
12-     public static String reverse(String s) throws Exception {
13-         if(s.length() == 0) {
14-             throw new Exception();
15-         }
16-         String reverseStr = "";
17-         for(int i = s.length() - 1; i >= 0; --i) {
18-             reverseStr += s.charAt(i);
19-         }
20-         return reverseStr;
21-     }
22- }
23- }
```

Analisa:

reverse melempar Exception jika string kosong. try-catch menangkap exception dan menampilkan pesan error, dan finally selalu menampilkan "All done".

Percobaan 10

```
1- import java.io.*;
2
3- class RandomAccessRevisi {
4-     public static void main(String[] args) {
5         String bookList[] = {"Satu", "Dua", "Tiga"};
6         int yearList[] = {1920, 1230, 1940};
7         try {
8             RandomAccessFile books = new RandomAccessFile("books.txt", "rw");
9             for (int i = 0; i < 3; i++) {
10                 books.writeUTF(bookList[i]);
11                 books.writeInt(yearList[i]);
12             }
13             books.seek(0);
14             System.out.println(books.readUTF() + " " + books.readInt());
15             System.out.println(books.readUTF() + " " + books.readInt());
16             books.close();
17         } catch (IOException e) {
18             System.out.println("Indeks melebihi batas");
19         }
20         System.out.println("test");
21     }
22 }
```

Analisa:
Program ini mencoba menulis dan membaca data menggunakan RandomAccessFile. Blok try-catch menangkap IOException yang mungkin terjadi selama operasi file. Jika ada kesalahan saat menulis atau membaca file, program akan menampilkan pesan "Indeks melebihi batas".

Percobaan 11

```
1- class RangeErrorException extends Throwable {
2-     public RangeErrorException(String s) {
3         super(s);
4     }
5
6-     public static void main(String[] args) {
7         int position = 1;
8         try {
9             if (position > 0) {
10                 throw new RangeErrorException("Position " + position);
11             }
12         } catch (RangeErrorException e) {
13             System.out.println("Range error: " + e.getMessage());
14         }
15         System.out.println("This is the last program.");
16     }
17 }
18 |
```

Analisa:
Program ini menunjukkan cara membuat dan menangani exception kustom yang extends Throwable. Ketika kondisi tertentu terpenuhi (position > 0), exception dilempar dan ditangkap dalam blok catch, yang kemudian menampilkan pesan error.

Percobaan 12

```
1- class MyException extends Exception {
2-     private String Teks;
3
4-     MyException(String s) {
5         Teks = "Exception generated by: " + s;
6         System.out.println(Teks);
7     }
8 }
9
10- class Eksepsi {
11-     static void tampil(String s) throws Exception {
12         System.out.println("Tampil");
13         if (s.equals("amir")) {
14             throw new MyException(s);
15         }
16         System.out.println("OK!");
17     }
18
19-     public static void main(String[] args) throws Exception {
20-         try {
21             tampil("ali");
22             tampil("amir");
23         } catch (MyException ex) {
24             System.out.println("Tangkap: " + ex);
25         }
26     }
27 }
```

Analisa:
Program ini menunjukkan bagaimana cara membuat dan menangani exception kustom yang extends Exception. Method tampil melempar MyException jika parameter yang diberikan adalah "amir". Blok try-catch di main menangkap MyException dan menampilkan pesan yang dihasilkan.