TUGAS & EVALUASI

Soal Tugas & Evaluasi

1. Jelaskan apa yang dimaksud dengan exception handling dan sebutkan tujujannya

Jawaban

Exception merupakan permasalaan yang terjadi Ketika mengeksekusi program

TUGAS & EVALUASI

Soal Tugas & Evaluasi

2. Sebutkan kategori exception handling beserta penjelasannya

Jawaban

1. Checked exception = exception yang terjadi pada saat (compile program time exceptions). Kategori ini tidak dapat dibiarkan begitu saja, kita harus menangani kesalahan ini jika tidak ingin program kita berhenti ditengah jalan
2. Unchecked exception = exception ini tidak harus ditangani seperti Checked exception. Sederhananya exception ini terjadi saat program menemukan sebuah bug seperti kesalahan logika program
3. User defined exception = dalam pemrograman java, programmer diperbolehkan membuat class exception sendiri. Ada beberapa untuk membuat class exception sendiri yaitu :
    a. Extends clas exceptions
    b. Implements interface throwable

TUGAS & EVALUASI

Soal Tugas & Evaluasi

3. buatlah kode program yang menunjukkan implementasi kategori exception handling berikut :

- checked exception = buat program yang menangani fileNotFoundException dan IOException
- unchecked exception = buat program yang menunjjukan penanganan ArithmeticException dan NullPointerException

Source Code

1.

```
import java.io.*;

public class CheckedException {
    public static void main(String[] args) {

        try {
            FileReader file = new
FileReader("filetidakada.txt");
            BufferedReader fileInput = new
BufferedReader(file);
            System.out.println(fileInput.readLine());
            fileInput.close();
        } catch (FileNotFoundException e) {
            System.out.println("Kesalahan: File tidak
ditemukan.");
        } catch (IOException e) {
            System.out.println("Kesalahan: Terjadi
IOException.");
        }


        try {
            FileWriter fileWriter = new
FileWriter("output.txt");
            fileWriter.write("Halo, dunia!");
            fileWriter.close();
        } catch (IOException e) {
            System.out.println("Kesalahan: Terjadi IOException
saat menulis.");
        }
    }
}
```

2.

```
package bab9No4;
```

TUGAS & EVALUASI

```java
public class UncheckedException {
    public static void main(String[] args) {

        try {
            int hasil = 1 / 0;
        } catch (ArithmeticException e) {
            System.out.println("Kesalahan: Tidak bisa membagi
dengan nol.");
        }


        try {
            String str = null;
            System.out.println(str.length());
        } catch (NullPointerException e) {
            System.out.println("Kesalahan: Mencoba mengakses
objek yang null.");
        }
    }
}
```

Penjelasan

Program ini menunjukkan cara menangani Unchecked Exception atau Runtime Exception di Java menggunakan blok try-catch. Unchecked exceptions adalah jenis kesalahan yang biasanya terjadi saat program berjalan (runtime) dan tidak diharuskan untuk ditangani secara eksplisit dalam kode, meskipun kita masih bisa menangani kesalahan tersebut jika diperlukan.

Pada program ini, ada dua contoh kesalahan yang umum terjadi dalam aplikasi Java:

Kesalahan Pembagian dengan Nol (ArithmeticException): Pada blok pertama, program mencoba melakukan pembagian dengan nol (1 / 0). Pembagian dengan nol adalah kesalahan aritmatika yang menyebabkan Java melemparkan sebuah ArithmeticException. Program menangkap kesalahan ini menggunakan catch dan mencetak pesan kesalahan: "Kesalahan: Tidak bisa membagi dengan nol."

TUGAS & EVALUASI

Kesalahan Null Pointer (NullPointerException): Pada blok kedua, program mencoba mengakses panjang dari sebuah string yang bernilai null. Karena objek str tidak mengarah ke data valid (nilai null), Java akan melemparkan NullPointerException. Program menangkap kesalahan ini dengan blok catch dan mencetak pesan kesalahan: "Kesalahan: Mencoba mengakses objek yang null."

Secara keseluruhan, program ini menunjukkan bagaimana cara mengelola kesalahan yang terjadi saat program berjalan, meskipun kesalahan tersebut adalah jenis exception yang tidak wajib ditangani (unchecked exception). Dengan menangkap dan menangani exception menggunakan try-catch, program dapat melanjutkan eksekusinya tanpa berhenti mendadak karena kesalahan yang terjadi.

Output

1.

```
Kesalahan: File tidak ditemukan.

Process finished with exit code 0
```

2.

```
Kesalahan: Tidak bisa membagi dengan nol.
Kesalahan: Mencoba mengakses objek yang null.

Process finished with exit code 0
```

TUGAS & EVALUASI

Soal Tugas & Evaluasi

4. Buat sebuah program yang melakukan operas pembagian angka dengan penanganan menggunakan blok try-catchfinally. Program ini harus dapat menangani input yang tidak valid, termasuk pembagian dengan nol

Source Code

```
package bab9No5;

import java.util.Scanner;

public class PembagianMemakaiTryCatchFinally {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Masukkan angka pembilang: ");
            double pembilang = scanner.nextDouble();

            System.out.print("Masukkan angka penyebut: ");
            double penyebut = scanner.nextDouble();

            if (penyebut == 0) {
                throw new ArithmeticException("Kesalahan:
Tidak bisa membagi dengan nol.");
            }

            double hasil = pembilang / penyebut;
            System.out.println("Hasil pembagian: " + hasil);
        } catch (ArithmeticException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("Kesalahan: Input tidak valid.
Harap masukkan angka.");
        } finally {
            System.out.println("Program selesai. Terima
kasih!");
            scanner.close();
        }
    }
}
```

Penjelasan

Program ini menunjukkan cara menangani **Checked Exception** di Java, yang merupakan jenis exception yang harus ditangani oleh programmer. Checked

exceptions terjadi saat program berinteraksi dengan sumber daya eksternal, seperti file atau jaringan, yang mungkin tidak tersedia atau mengalami kesalahan.

Pada blok pertama, program mencoba membuka dan membaca file bernama `filetidakada.txt` menggunakan `FileReader` dan `BufferedReader`. Jika file tidak ditemukan, Java akan melemparkan `FileNotFoundException`, yang merupakan turunan dari `IOException`. Program menangkap exception ini dengan `catch` dan mencetak pesan "Kesalahan: File tidak ditemukan." Jika terjadi kesalahan lain yang berhubungan dengan input/output, seperti masalah saat membaca file, `IOException` akan ditangkap dan mencetak pesan "Kesalahan: Terjadi IOException."

Pada blok kedua, program mencoba menulis teks "Halo, dunia!" ke dalam file `output.txt` menggunakan `FileWriter`. Jika terjadi kesalahan saat menulis, seperti masalah dengan akses file atau penyimpanan, `IOException` akan ditangkap dan mencetak pesan "Kesalahan: Terjadi IOException saat menulis."

Dengan menangani **Checked Exception** menggunakan blok `try-catch`, program dapat mencegah aplikasi berhenti mendadak dan memberikan pesan yang informatif kepada pengguna jika terjadi masalah dengan file atau input/output.

Output

```
Masukkan angka pembilang: 10
Masukkan angka penyebut: 2
Hasil pembagian: 5.0
Program selesai. Terima kasih!
```