

LAPORAN PRAKTIKUM

SESI #11

Execption Pemograman generik dan design pattern



Nama: Andre Saputra

Npm : 23753045

Kelas: MI 3B

PROGRAM STUDI MANAJEMEN INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI LAMPUNG

2024

Minggu ke 11

Unit Kompetensi :Exception Pemograman generik dan design pattern

Waktu : 340 menit

Tempat : Lab Software

1. Elemen Kompetensi

- Exception pemograman generik dan design pattern

2. Indikator Kinerja

- Mampu menjelaskan exception pemograman generik dan design pattern
- Mampu menjelaskan syntax-syntax untuk membuat exception pemograman generik dan design pattern

3. Teori

dasarnya ada tiga jenis error didalam pemrograman dengan Java, yaitu: syntax error, logic error, dan runtime error. Syntax error merupakan kesalahan yang terjadi saat programmer salah menuliskan kode program. Misalnya, saat programmer kurang memberikan titik koma di akhir dari statement. Biasanya error ini akan muncul saat progam di-compile. Logic error berkaitan dengan kesalahan yang dilakukan programmer saat programmer membuat kode sehingga akan menghasilkan output kode yang tidak sesuai dengan yang diinginkan. Kesalahan logic error seringkali disebut sebagai bug/kutu. Untuk menangani bug dilakukan proses debugging. Proses debugging adalah mencari satu persatu baris perbaris kode yang mana yang membuat output yang dihasilkan tidak sesuai

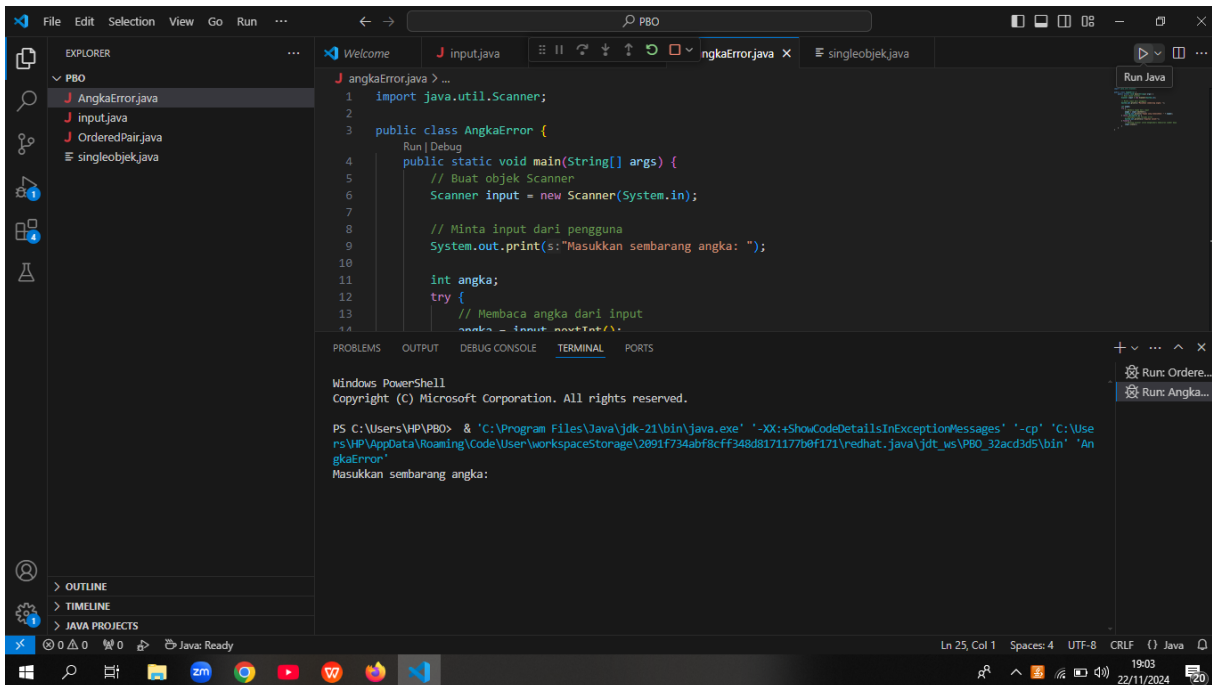
4. Pada Bahan dan alat

- Laptop
- XAMPP
- Visual Studio Code
- Browser

5. Organisasi

- Mandiri

6. Hasil praktikum



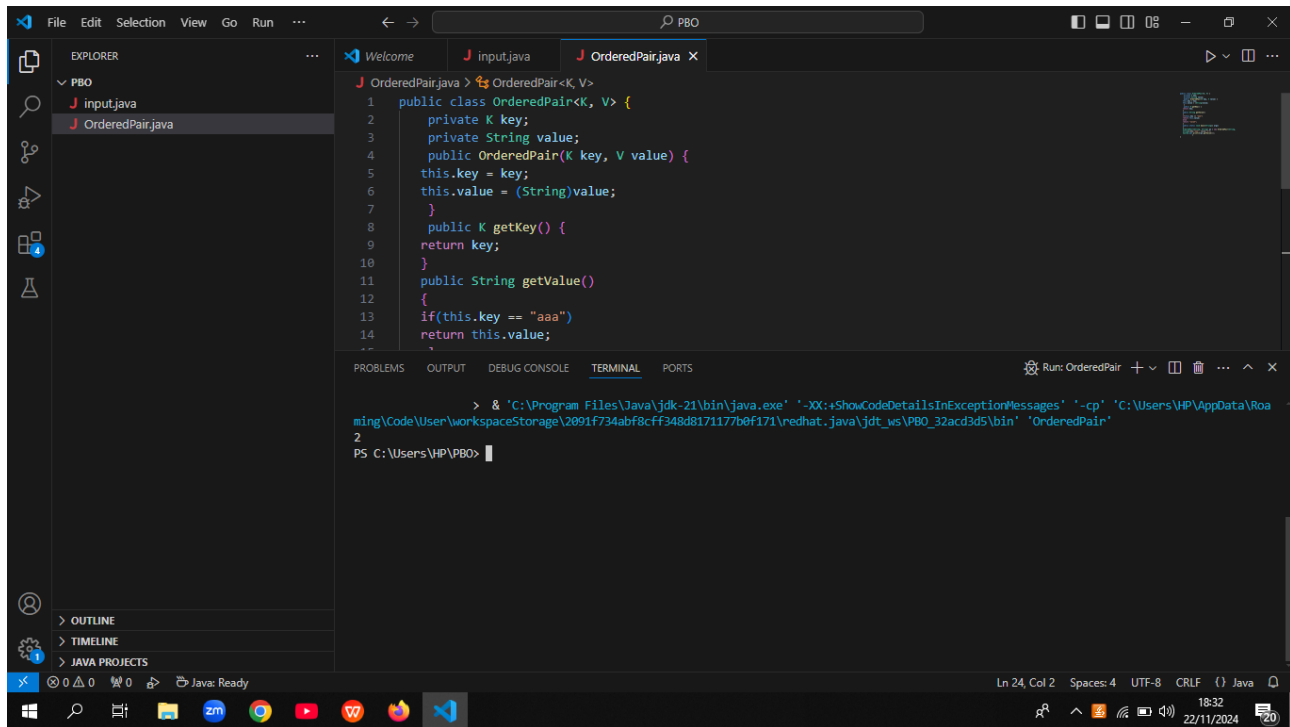
```
1 import java.util.Scanner;
2
3 public class AngkaError {
4     public static void main(String[] args) {
5         // Buat objek Scanner
6         Scanner input = new Scanner(System.in);
7
8         // Minta input dari pengguna
9         System.out.print(s:"Masukkan sembarang angka: ");
10
11         int angka;
12         try {
13             // Membaca angka dari input
14             angka = input.nextInt();
15         } catch (Exception e) {
16             // Tangani error jika input bukan angka
17             System.out.println("Error: Input bukan angka.");
18         }
19     }
20 }
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

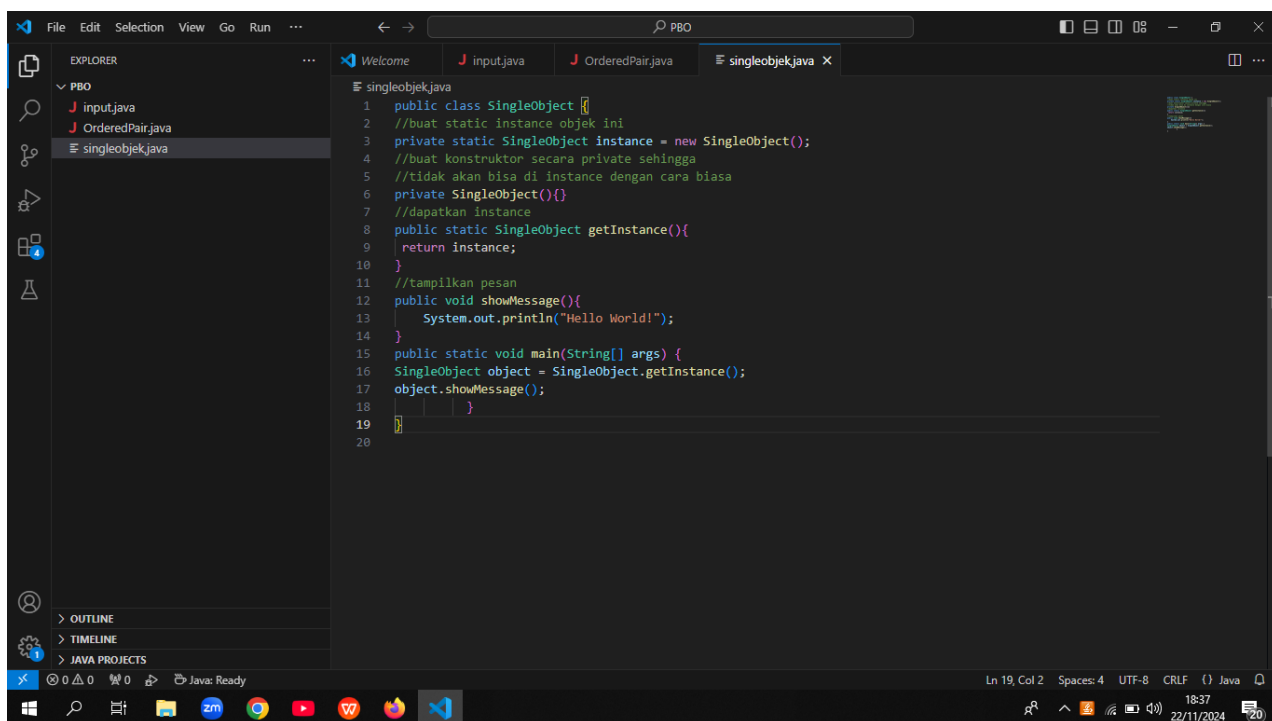
PS C:\Users\VP\Documents> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\VP\AppData\Roaming\Code\User\workspaceStorage\2091f734bf8c8ff348d8171177b0f171\redhat\.java\jdt_ws\p80_32acd3d5\bin' 'AngkaError'

Masukkan sembarang angka:

Angka error



Orderredpair.java



1. Jelaskan bagaimana cara menangani exception?

Jawab; • **Blok try**: Tempat kode yang berpotensi menimbulkan error atau exception.

• **Blok catch**: Menangkap dan menangani exception jika terjadi dalam blok try.

2. . Sebutkan macam-macam exception di Java (10 saja)?

1. Jawab; **ArithmeticException**: Terjadi saat operasi aritmatika yang ilegal, seperti pembagian dengan nol.
2. **NullPointerException**: Terjadi saat mencoba mengakses objek yang belum diinisialisasi (null).
3. **ArrayIndexOutOfBoundsException**: Terjadi saat mencoba mengakses indeks array yang tidak valid.
4. **ClassNotFoundException**: Terjadi ketika kelas yang diminta tidak ditemukan.
5. **FileNotFoundException**: Terjadi ketika file yang diminta tidak ditemukan.
6. **IOException**: Terjadi ketika terjadi kesalahan input/output.
7. **NumberFormatException**: Terjadi saat mencoba mengonversi string ke angka yang tidak valid.
8. **IllegalArgumentException**: Terjadi ketika argumen yang diberikan tidak sesuai atau tidak valid.
9. **IndexOutOfBoundsException**: Terjadi saat mencoba mengakses indeks yang tidak valid pada koleksi seperti list atau array.
10. **InterruptedException**: Terjadi ketika sebuah thread terinterupsi selama eksekusi.

3. Apakah class generik bisa diwariskan?

Jawab; Ya, **class generik** bisa diwariskan di Java. Sebuah class generik dapat mewariskan class generik lainnya, dan subclass-nya juga bisa mendeklarasikan tipe parameter generiknya

4. Apakah design pattern singleton dapat diwariskan?

Jawab; Alasan utamanya adalah bahwa pattern Singleton dirancang untuk memastikan hanya ada satu instance dari kelas tersebut. Jika kelas Singleton diwariskan, maka subclass dapat membuat instance terpisah, yang bertentangan dengan prinsip dasar Singleton (hanya satu instance).

Namun, Anda dapat menggunakan **composition** untuk mencapai tujuan serupa dengan mewariskan perilaku tertentu dari superclass, tetapi menjaga pola Singleton tetap berlaku di kelas utama.

