

## Worksheet Pertemuan 14 Fundamen Pengembangan Aplikasi Exception Handling

Nama : Danendra Farrel Adriansyah  
NIM : 23523170

### A. Membuat Folder untuk Menyimpan Hasil Praktikum

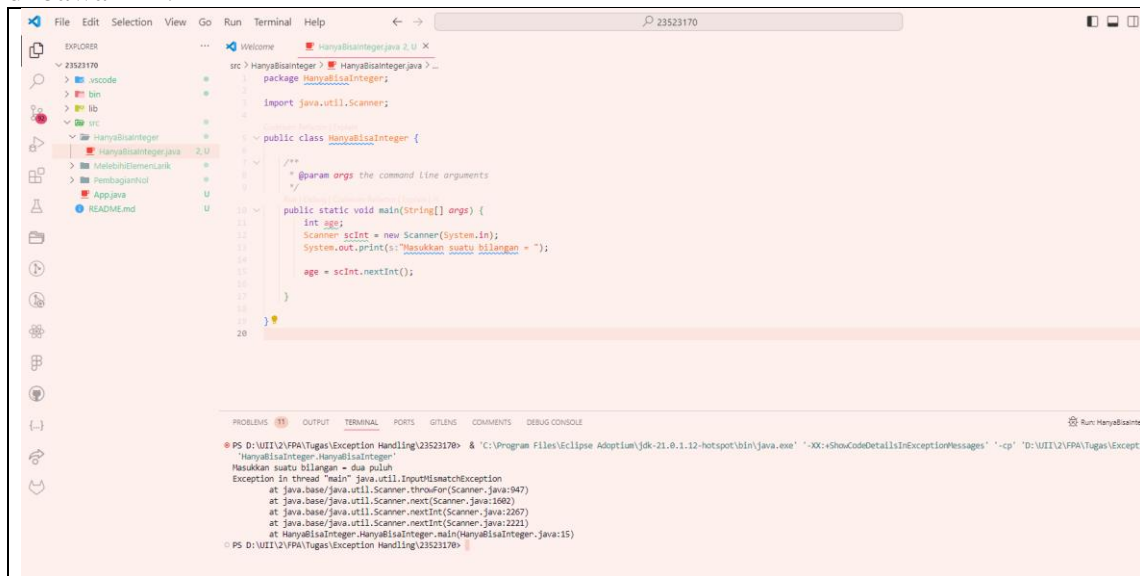
1. Siapkan **proyek VS Code** dengan nama menggunakan **NIM** masing-masing. Jika proyek NIM masih memuat hasil praktikum sebelumnya dan akan dimanfaatkan lagi, jangan lupa pindahkan dulu isinya ke folder lain sebagai arsip.
2. Folder ini akan dijadikan tempat untuk menyimpan hasil praktikum materi *exception handling* ini.

### B. Pesan Kesalahan Ketika Variabel Diisi dengan Nilai yang Berbeda Tipe Data

1. Ekstrak file **HanyaBisaInteger.zip** sebagai **package** pada proyek NIM.
2. Jalankan program tersebut kemudian ketika muncul permintaan untuk memasukkan bilangan bulat, isikan tulisan "**dua puluh**".
3. Jelaskan secara singkat apa yang terjadi!

InputMismatchException karena inputan kita string, sedangkan yang diminta integer (scInt), int age.

4. Setelah Anda jawab, salin tangkapan layar (SS) dari jendela Output VS Code ke kotak di bawah ini:



```
File Edit Selection View Go Run Terminal Help
23523170
Welcome
src > HanyaBisaInteger > HanyaBisaInteger.java > ...
package HanyaBisaInteger;
import java.util.Scanner;

public class HanyaBisaInteger {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int age;
        Scanner scInt = new Scanner(System.in);
        System.out.print("Masukkan suatu bilangan = ");
        age = scInt.nextInt();
    }
}

PROBLEMS OUTPUT TERMINAL PORTS GIT LENS COMMENTS DEBUG CONSOLE
PS D:\UIT\2\VFPA\Tugas\Exception Handling\23523170> & "C:\Program Files\Eclipse Adoptium\jdk-21.0.1-12-hotspot\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -cp "D:\UIT\2\VFPA\Tugas\Except
"HanyaBisaInteger.HanyaBisaInteger"
Masukkan suatu bilangan = dua puluh
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:947)
    at java.base/java.util.Scanner.next(Scanner.java:1602)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2267)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2221)
    at HanyaBisaInteger.HanyaBisaInteger.main(HanyaBisaInteger.java:15)
PS D:\UIT\2\VFPA\Tugas\Exception Handling\23523170>
```

Perhatikan kata-kata pada baris pertama untuk menjawab butir C!

### C. Menangkap Nama Exception

1. Setelah Anda jalankan instruksi pada butir B dan Anda jawab, tulislah kata terakhir yang muncul pada baris pertama berwarna merah di jendela OUTPUT (di bawah tulisan **Masukkan suatu bilangan = dua puluh**, setelah )

java.util.InputMismatchException

2. Itulah nama *exception* untuk menangani kesalahan tipe data yang dimasukkan. Karena menyangkut nama, *tuliskan persis seperti ada yang tertera di baris pertama merah itu yaa* (huruf besar dan kecil harus diperhatikan)
3. **Ingat!** bukan satu baris utuh yaa, tapi satu kata panjang pada "**java.util.??????**"

#### D. Penanganan Kesalahan Masukan

1. Carilah baris program pada **main class** yang digunakan untuk membaca umur (*age*) seperti di bawah ini

```
age = scInt.nextInt();
```

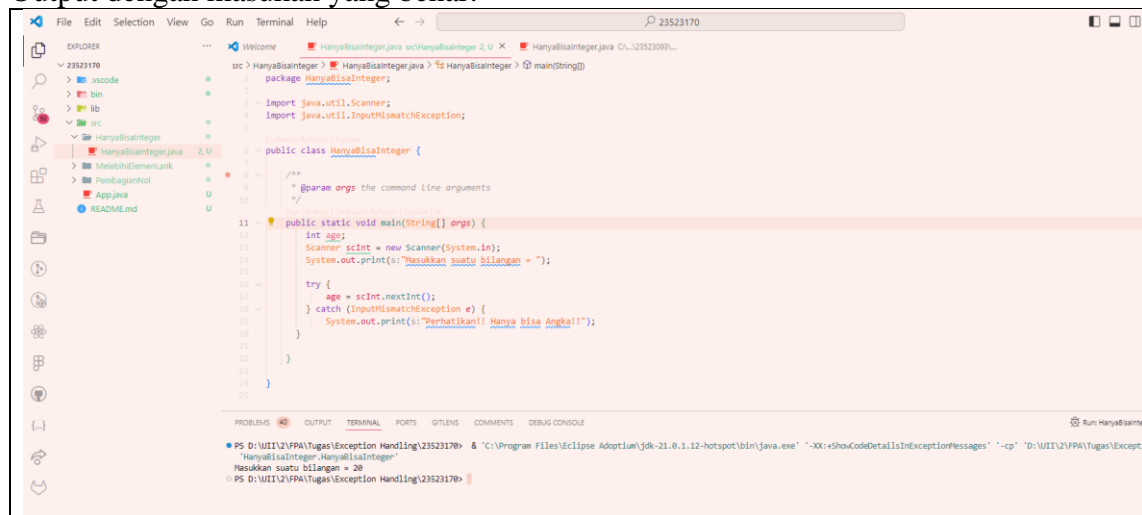
diganti dengan enam baris perintah berikut:

**Awat!** ada 2 komentar yang harus Anda jalankan!

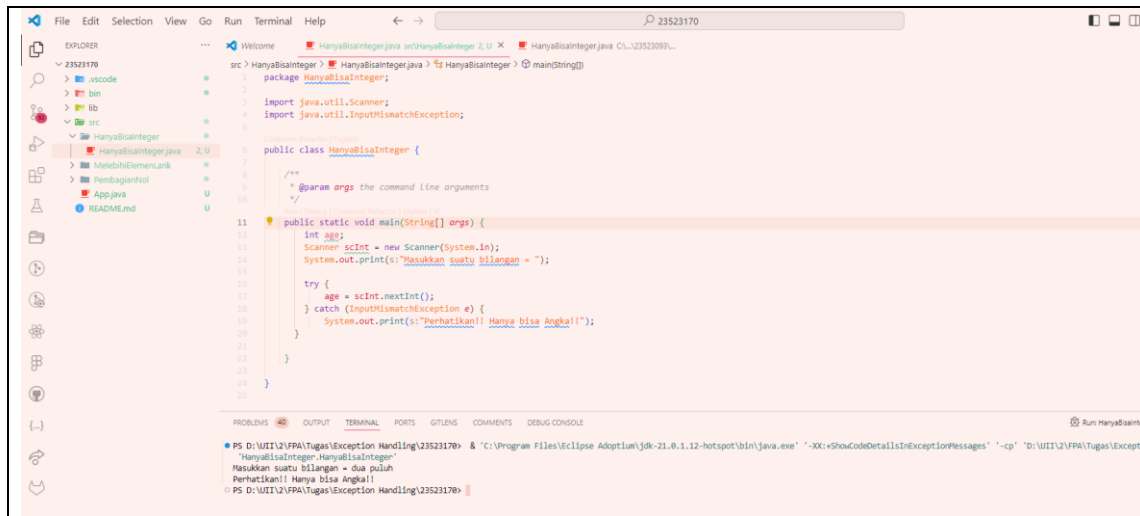
```
try {
    age = scInt.nextInt();
}
/// ganti AWASINIDIGANTI dengan isian pada butir C
catch (AWASINIDIGANTI e) {
//// ganti INIJUGADIGANTI dengan kata-kata Anda sendiri
    System.out.print("INIJUGADIGANTI");
}
```

2. Setelah benar-benar tidak ada kesalahan, jalankan program Anda 2x, yang pertama berilah nilai *variable* umur dengan benar & yang kedua berilah nilai **variable** umur dengan cara yang salah, kemudian salinlah *screenshot* dari jendela *output*.

Output dengan masukan yang benar:



Output dengan masukan yang salah:



```
File Edit Selection View Go Run Terminal Help
src > HanyaBisaInteger > HanyaBisaInteger.java > HanyaBisaInteger > main(String[])
package HanyaBisaInteger;

import java.util.Scanner;
import java.util.InputMismatchException;

public class HanyaBisaInteger {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int age;
        Scanner scInt = new Scanner(System.in);
        System.out.print("Masukkan suatu bilangan = ");

        try {
            age = scInt.nextInt();
        } catch (InputMismatchException e) {
            System.out.print("Perhatikan!! Hanya bisa Angka!");
        }
    }
}
```

PROBLEMS OUTPUT TERMINAL PORTS GITLENS COMMENTS DEBUG CONSOLE

PS D:\UIT\2\FPA\Tugas\Exception Handling\23523170> & "C:\Program Files\Eclipse Adoptium\jdk-21.0.1.12-hotspot\bin\java.exe" "-Xmx<showCodeDetailsInExceptionMessages" "-cp" "D:\UIT\2\FPA\Tugas\Except  
HanyaBisaInteger> HanyaBisaInteger  
Masukkan suatu bilangan = dua puluh  
Perhatikan!! Hanya bisa Angka!  
PS D:\UIT\2\FPA\Tugas\Exception Handling\23523170>

### E. Penanganan Kesalahan yang Disebabkan oleh Pembagian Bilangan Nol

1. Ekstrak file **PembagianNol.zip** sebagai *package* pada proyek NIM.
2. Proyek ini adalah perhitungan rumus kecepatan, yang mengandung pembagian jarak dengan waktu.
3. Ketika program dijalankan, isikan data untuk *variable waktu* dengan **bilangan nol**. Pasti **error** kan?
4. Coba atasi kesalahan ini dengan cara seperti yang sudah dicoba pada butir B sampai butir D.
5. **Ingat!** kuncinya di sini adalah kata terakhir pada baris pertama. Pasti yang muncul akan berbeda dengan pada kasus kesalahan tipe data masukan.
6. Jangan lupa beri peringatan yang sesuai dengan kesalahan yang sedang ditangani.
7. Anda harus buat rumus sendiri yaa, jangan sampai sama dengan punya teman Anda.
8. Simpan proyek Anda.

### F. Penanganan Kesalahan yang Disebabkan oleh Indeks Larik Melebihi Jangkauan Elemen

1. Ekstrak file **MelebihiElemenLarik.zip** sebagai *package* pada proyek NIM.
2. Proyek ini mengandung larik dengan elemen sebanyak 10.
3. *Run* program tersebut, dan ketika diminta mengisi pada bagian “*Masukkan indeks elemen yang akan dibaca =*”, isilah dengan bilangan yang **lebih dari 10**, misal 100, 200 atau berapapun **asal lebih dari 10**.
4. Sebagaimana praktikum sebelumnya, modifikasi proyek Anda ini agar terdapat pesan ketika pengguna memasukkan elemen larik yang lebih dari jangkauannya.
5. Simpan proyek Anda.

- Kesalahan-kesalahan lain yang terjadi ketika program di-run (*runtime error*) dapat diatasi sebagaimana contoh-contoh yang dipraktikkan ini.
- Kompres folder NIM sebagai file ZIP dan simpan file ini sebagai file pdf.
- Submit ke classroom atau ruang pengumpulan lainnya.