

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN BERBASIS OBJEK**  
**PERTEMUAN 8**



Disusun oleh :

Nama : Fatimah Az Zahra

NIM : 2211102160

Kelas : IF-10-K

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**  
**2024**

## MODUL 08

### A. Guided

#### 1. TryCatch1.java

##### Source Code

```
5 package GUIDED1; // Menyatakan bahwa kelas ini berada di dalam paket GUIDED1
6 // Fatimah Az Zahra-2211102160-IF10K
7
8 public class TryCatch1 {
9
10     public static void main(String[] args) {
11         try { // Memulai blok try untuk menangani kode yang mungkin menghasilkan exception
12             int[] arr = new int[1]; // Mendeklarasikan dan menginisialisasi array dengan 1 elemen
13             System.out.println(arr[1]); // Mencoba mengakses elemen kedua dari array yang tidak ada
14             System.out.println(
15                 "Baris ini tidak akan dieksekusi, karena statement baris diatas terjadi exception");
16             // Baris ini tidak akan dieksekusi karena akan terjadi exception pada baris sebelumnya
17         } catch (ArrayIndexOutOfBoundsException e) { // Menangkap exception jika terjadi
18             System.out.println("Terjadi exception karena indeks di luar kapasitas array");
19             // Pesan yang ditampilkan jika ada exception ArrayIndexOutOfBoundsException
20         }
21         System.out.println("Setelah blok try catch");
22         // Baris ini akan dieksekusi setelah blok try-catch, terlepas dari apakah terjadi exception atau tidak
23     }
24 }
```

##### Output:

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul8 ---
Terjadi exception karena indeks di luar kapasitas array
Setelah blok try catch

-----
BUILD SUCCESS
-----

Total time: 3.897 s
Finished at: 2024-05-26T14:47:38+07:00
-----
```

##### Penjelasan:

Program `TryCatch1` mendemonstrasikan penanganan exception menggunakan blok `try-catch`. Program ini dimulai dengan mendeklarasikan dan menginisialisasi array `arr` berukuran satu elemen. Selanjutnya, program mencoba mengakses elemen kedua dari array yang tidak ada (indeks 1), menyebabkan `ArrayIndexOutOfBoundsException`. Exception ini kemudian ditangkap oleh blok `catch`, yang menampilkan pesan kesalahan "Terjadi exception karena indeks di luar kapasitas array". Setelah itu, eksekusi dilanjutkan dan menampilkan pesan "Setelah blok try catch". Program ini menunjukkan bagaimana exception ditangani agar program tidak terhenti secara tidak terduga.

## TryCatch2.Java

### Source Code

```
5 package GUIDED1;
6 // Fatimah Az Zahra-2211102160-IF10K
7 public class TryCatch2 {
8
9     public static void main(String[] args) {
10         try {
11             int x = args.length; //karena ga ada nilainya otomatis program membaca x=0
12             int y = 100 / x; //semua bilangan yang dibagi 0 hasilnya tidak terdefinisi
13             int[] arr = {10, 11}; //ga guna jadinya karena ambigu ini x atau y
14             y = arr[x]; //nampilin x tapi arr x ga diketahui
15             System.out.println("Tidak terjadi exception");
16         } catch (ArithmeticException e) {
17             System.out.println("Terjadi exeption karena pembagian dengan nol");
18         } catch (ArrayIndexOutOfBoundsException e) {
19             System.out.println("Terjadi exeption karena indeks di luar kapasitas array");
20         }
21         System.out.println("Setelah blok try catch");
22     }
23 }
```

### Output :

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul8 ---
Terjadi exeption karena pembagian dengan nol
Setelah blok try catch

-----
BUILD SUCCESS
-----

Total time: 1.878 s
Finished at: 2024-05-26T14:49:34+07:00
-----
```

### Penjelasan program:

Pada `TryCatch2`, program menginisialisasi variabel `x` dengan panjang argumen yang diberikan (0 jika tidak ada argumen). Kemudian, program mencoba melakukan pembagian 100 dengan `x`, yang menyebabkan `ArithmeticException` karena pembagian dengan nol. Jika argumen ada, program mencoba mengakses elemen array `arr` dengan indeks `x`, yang dapat menyebabkan `ArrayIndexOutOfBoundsException` jika `x` melebihi panjang array. Exception yang ditangkap akan menampilkan pesan sesuai dengan jenis kesalahannya. Setelah penanganan, program melanjutkan dan menampilkan pesan "Setelah blok try catch".

## 2. Throw.Java

### Source Code

```
5 package GUIDED2;
6 //Fatimah Az Zahra-2211102160-IF10K
7
8 public class Throw {
9
10     public static void methodLain() {
11         try {
12             throw new ArrayIndexOutOfBoundsException(1);
13         } catch (ArrayIndexOutOfBoundsException e) { // e bisa diganti apapun, disesuaikan aja
14             System.out.println("Penanganan exception dalam method methodLain()");
15             throw e;
16         }
17     }
18
19     //metode yang ada kemungkinan error di main di lempar ke try catch
20     public static void main(String[] args) {
21         try {
22             methodLain();
23         } catch (ArrayIndexOutOfBoundsException e) {
24             System.out.println("Penanganan exception dalam method main()");
25         }
26     }
27 }
```

### Output:

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul8 ---
Penanganan exception dalam method methodLain()
Penanganan exception dalam method main()
-----
BUILD SUCCESS
-----
Total time: 2.320 s
Finished at: 2024-05-26T14:51:13+07:00
-----
```

### Penjelasan program:

Program `Throw` dalam paket `GUIDED2` mendemonstrasikan penggunaan exception yang dilempar kembali (`rethrowing`) dari satu metode ke metode lain. Program ini memiliki dua metode: `methodLain` dan `main`. Pada `methodLain`, sebuah `ArrayIndexOutOfBoundsException` sengaja dilempar menggunakan perintah `throw`. Exception ini kemudian ditangkap oleh blok `catch` dalam metode yang sama, yang mencetak pesan "Penanganan exception dalam method methodLain()", dan setelah itu exception dilempar kembali menggunakan `throw e`.

Metode `main` memanggil `methodLain` dalam blok `try`. Ketika `methodLain` melempar kembali exception, exception tersebut ditangkap oleh blok `catch` di `main`, yang mencetak pesan "Penanganan exception dalam method main()".

### Output program adalah:

```
Penanganan exception dalam method methodLain()
Penanganan exception dalam method main()
Setelah blok try catch
```

Output ini menunjukkan bahwa exception ditangani pertama kali di `methodLain`, kemudian dilempar kembali dan ditangani lagi di `main`, memastikan bahwa program tidak terhenti secara tidak terduga dan semua exception yang terjadi ditangani dengan baik.

### 3. DemoFinally.java

#### Source Code

```
5 package GUIDED3;
6 //Fatimah Az Zahra-2211102160-IF10K
7
8 public class DemoFinally {
9
10     public static void main(String[] args) {
11         int x = 3; //urutan indeks ke 3
12         int[] arr = {10, 11, 12}; //array mulai dari 0
13         try{
14             System.out.println(arr[x]);
15             System.out.println("Tidak terjadi exception");
16         } catch (ArrayIndexOutOfBoundsException e) {
17             System.out.println("Terjadi exception");
18             System.out.println("indeks diluar batas: " + e.getMessage());
19         } finally {
20             System.out.println("Program Selesai");
21         }
22     }
23 }
```

#### Output:

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul8 ---
Terjadi exception
indeks diluar batas: Index 3 out of bounds for length 3
Program Selesai

-----
BUILD SUCCESS
-----

Total time: 2.168 s
Finished at: 2024-05-26T14:56:02+07:00
-----
```

#### Penjelasan

Program `DemoFinally` dalam paket `GUIDED3` menunjukkan penggunaan blok `finally` yang akan dijalankan terlepas dari apakah terjadi exception atau tidak. Program ini mendeklarasikan sebuah array `arr` dengan elemen-elemen `{10, 11, 12}` dan sebuah variabel `x` dengan nilai 3. Array memiliki indeks mulai dari 0 hingga 2, sehingga mengakses `arr[3]` akan menyebabkan `ArrayIndexOutOfBoundsException`.

Dalam blok `try`, program mencoba mencetak elemen ke-3 dari array yang menyebabkan exception. Ketika exception terjadi, blok `catch` menangkap `ArrayIndexOutOfBoundsException` dan mencetak pesan "Terjadi exception" serta pesan tambahan yang menunjukkan indeks yang diakses berada di luar batas array. Setelah blok `catch` selesai, blok `finally` dieksekusi, mencetak pesan "Program Selesai".

Output dari program ini adalah:

```
Terjadi exception
indeks diluar batas: Index 3 out of bounds for length 3
Program Selesai
```

#### Penjelasan output:

1. `Terjadi exception`: Dicitak oleh blok `catch` karena terjadi `ArrayIndexOutOfBoundsException`.

2. `indeks diluar batas: Index 3 out of bounds for length 3`:  
Memberikan informasi detail tentang exception yang terjadi.

3. `Program Selesai`: Dicitak oleh blok `finally`, menunjukkan bahwa blok `finally` dieksekusi meskipun terjadi exception.

Blok `finally` memastikan bahwa pesan "Program Selesai" selalu dicetak, menandakan akhir dari program.

#### 4. DemoStream1.java

##### Source Code

```
5 package GUIDED4;
6 //Fatimah Az Zahra-2211102160-IF10K
7
8 import java.io.*;
9
10 public class DemoStream1 {
11
12     public static void main(String[] args) {
13         byte[] data = new byte[10]; //menampilkan 10 karakter, lebih dari 10 tidak dapat ditampilkan
14         System.out.print("Masukkan data : "); //input
15         try {
16             System.in.read(data);
17         } catch (IOException e) {
18             System.out.print("Terjadi Exception");
19         }
20         System.out.print("Yang anda ketik : "); //output
21         for (int i = 0; i < data.length; i++) {
22             System.out.print((char) data[i]);
23             // (char) diatas disebut casting yakni untuk mengubah format menjadi char
24         }
25     }
26 }
```

##### Output:

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul8 ---
Masukkan data : Fatimah
Yang anda ketik : Fatimah

-----
BUILD SUCCESS
-----

Total time: 14.459 s
Finished at: 2024-05-26T14:58:24+07:00
-----
```

##### Penjelasan:

Program `DemoStream1` dalam paket `GUIDED4` mendemonstrasikan penggunaan aliran (stream) untuk membaca input dari pengguna menggunakan `System.in` dan menampilkannya kembali sebagai output. Program ini dimulai dengan mendeklarasikan array byte `data` dengan panjang 10, yang akan menyimpan 10 karakter input dari pengguna. Program kemudian meminta pengguna untuk memasukkan data melalui prompt "Masukkan data :".

Dalam blok `try`, program mencoba membaca input dari pengguna menggunakan `System.in.read(data)`. Jika terjadi kesalahan I/O (IOException) selama pembacaan, blok `catch` akan menangkap exception dan mencetak pesan "Terjadi Exception".

Setelah pembacaan input, program menampilkan pesan "Yang anda ketik :". Kemudian, program menggunakan loop `for` untuk mengiterasi setiap elemen dalam array `data` dan mencetaknya sebagai karakter dengan melakukan casting dari byte ke char `(char) data[i]`.

Misalkan pengguna memasukkan "Fatimah" diikuti dengan menekan Enter, outputnya akan menjadi:

---

```
Masukkan data : Fatimah
Yang anda ketik : Fatimah
,,
```

### Penjelasan Output:

1. "Masukkan data : ": Program meminta pengguna untuk memasukkan data.
2. Input Pengguna: Misalnya, pengguna mengetik "Fatimah" dan menekan Enter.
3. "Yang anda ketik : ": Program menampilkan teks ini sebelum mencetak input pengguna.
4. "Fatimah": Program mencetak input yang telah dibaca dari pengguna. Karena array `data` berukuran 10 byte, input lebih dari 10 karakter hanya akan menyimpan 10 karakter pertama.

Jika pengguna mengetik lebih dari 10 karakter, misalnya "Fatimah Az Zahra", hanya 10 karakter pertama ("Fatimah A") yang akan ditampilkan. Karakter ke-10 akan menjadi newline (Enter) karena `System.in.read` juga menangkap karakter newline sebagai bagian dari input.

### DemoStream2.java

#### Source Code

```
5 package GUIDED4;
6 //Fatimah Az Zahra-2211102160-IF10K
7 import java.io.*;
8
9 public class DemoStream2 {
10
11     public static void main(String[] args) {
12         byte[] data = new byte[10];
13         int panjang = 0;
14         System.out.print("Masukkan data : ");
15         try {
16             panjang = System.in.read(data);
17             //Sistem.in.read mengembalikan panjang karakter yang
18             //diinputkan (termasuk enter yang dianggap 2 karakter..)
19             System.out.print("Yang anda ketik : ");
20             System.out.write(data);
21             System.out.println("Panjang Karakter : " + panjang);
22             System.out.print("index ke-1 sebnyk 3 : ");
23             System.out.write(data, 1, 3);
24         } catch (IOException e) {
25             System.out.print("Terjadi Exception");
26         }
27     }
28 }
29
30 /* write mencetak apapun tipe data yang ada, sedangkan print dan println mencetak data ke
    dalam tipe string */
```

#### Output:

```

--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul8 ---
Masukkan data : Fatimah
Yang anda ketik : Fatimah
Panjang Karakter : 8
index ke-1 sebnk 3 : ati
-----
BUILD SUCCESS
-----
Total time: 7.427 s
Finished at: 2024-05-26T14:59:52+07:00
-----

```

### Penjelasan:

Program `DemoStream2` dalam paket `GUIDED4` mendemonstrasikan penggunaan aliran (stream) untuk membaca input dari pengguna dan menampilkannya kembali menggunakan metode `System.out.write`. Program dimulai dengan mendeklarasikan array byte `data` berukuran 10 untuk menyimpan input, serta variabel `panjang` untuk menyimpan panjang input yang dibaca. Setelah meminta pengguna untuk memasukkan data melalui prompt "Masukkan data :", program membaca input tersebut dan menyimpannya ke dalam array `data`, serta mengembalikan panjang input yang dibaca (termasuk karakter newline yang dianggap sebagai dua karakter).

Kemudian, program menampilkan pesan "Yang anda ketik :" diikuti dengan mencetak input yang dibaca dari pengguna. Selanjutnya, program mencetak panjang input yang dibaca dengan pesan "Panjang Karakter :". Terakhir, program mencetak tiga karakter mulai dari indeks ke-1 dari array `data` dengan pesan "index ke-1 sebnk 3 :".

### Penjelasan Output:

Jika pengguna memasukkan "Fatimah" diikuti dengan menekan Enter, outputnya akan menjadi:

```

'''
Masukkan data : Fatimah
Yang anda ketik : Fatimah
Panjang Karakter : 8
index ke-1 sebnk 3 : ati
'''

```

1. "Masukkan data : ":
  - Program meminta pengguna untuk memasukkan data.
  - Pengguna mengetik "Fatimah" dan menekan Enter.
2. "Yang anda ketik : Fatimah":
  - Program mencetak string yang telah dibentuk dari input pengguna.
3. "Panjang Karakter : 8":
  - Program mencetak panjang input yang dibaca. Panjang input sebenarnya adalah 7 karakter ("Fatimah") ditambah 1 karakter newline, total menjadi 8.
4. "index ke-1 sebnk 3 : ":
  - Program menampilkan pesan ini sebelum mencetak tiga karakter mulai dari indeks ke-1.
5. "ati":



- Program mencetak tiga karakter mulai dari indeks ke-1 dalam array ``data``, yaitu "ati" dari "Fatimah".

Dengan demikian, output program ini menunjukkan bagaimana input dari pengguna dibaca, panjang input dihitung, dan sebagian input ditampilkan kembali sesuai instruksi yang diberikan dalam program.

### DemoStream3.java

#### Source Code

```

5  package GUIDED4;
6      //Fatimah Az Zahra-2211102160-IF10K
7
8  import java.io.*;
9
10 public class DemoStream3 {
11
12     public static void main(String[] args) throws IOException {
13         char data; //character stream
14         String str = "";
15         BufferedReader buff
16             = new BufferedReader(new InputStreamReader(System.in));
17         System.out.print("Ketik : ");
18         data = (char) buff.read();
19         while (data != '\n') { //'\\n' artinya selesai, kalau r read aja
20             str += data;
21             data = (char) buff.read();
22         } //GA ADA END '\\r'
23         System.out.println("Yang diketik : " + str);
24         System.out.println("Program Selesai");
25     }
26 }

```

#### Output:

```

--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul8 ---
Ketik : Zahra
Yang diketik : Zahra
Program Selesai

-----
BUILD SUCCESS
-----

Total time:  19.015 s
Finished at: 2024-05-26T15:00:51+07:00
-----

```

#### Penjelasan:

Program ``DemoStream3`` dalam paket ``GUIDED4`` menunjukkan penggunaan ``BufferedReader`` untuk membaca input karakter dari pengguna melalui ``System.in``. Setelah meminta pengguna untuk memasukkan teks melalui prompt "Ketik :", program membaca karakter satu per satu dan membentuk sebuah string dari input pengguna hingga menekan tombol Enter.

Proses dimulai dengan inisialisasi variabel ``data`` untuk menyimpan karakter yang dibaca, dan variabel ``str`` untuk menyimpan string yang akan dibentuk. Selanjutnya, program membaca karakter pertama menggunakan ``buff.read()`` dan memasukkannya ke dalam variabel ``data``. Proses pembacaan karakter dilakukan dalam loop ``while``, yang berlangsung selama karakter yang dibaca

ditambahkan ke dalam variabel `str` untuk membentuk string akhir.

Output dari program ini bergantung pada input yang dimasukkan oleh pengguna.

Jika pengguna memasukkan "zahra" diikuti dengan menekan Enter, outputnya akan menjadi:

```
```
```

```
Ketik : zahra
```

bukan merupakan karakter newline (`'\n'`). Setiap karakter yang dibaca

```
Yang diketik : zahra
```

```
Program Selesai
```

```
```
```

1. "Ketik : ":
  - Program meminta pengguna untuk memasukkan teks.
2. Input Pengguna:
  - Pengguna mengetik "zahra" dan menekan Enter.
3. "Yang diketik : zahra":
  - Program mencetak string yang telah dibentuk dari input pengguna.
4. "Program Selesai":
  - Program menampilkan pesan bahwa program telah selesai.

Output tersebut menunjukkan bahwa program berhasil membaca input "zahra" yang dimasukkan oleh pengguna dan mencetaknya kembali. Setelah itu, program menampilkan pesan bahwa proses sudah selesai.

## DemoStream4.java

### Source Code

```
5 package GUIDED4;
6 //Fatimah Az Zahra-2211102160-IF10K
7
8 import java.io.*;
9
10 public class DemoStream4 {
11
12     public static void main(String[] args) {
13         byte[] data = new byte[10];
14         int panjang = 0;
15         System.out.print("Masukkan data : ");
16         try {
17             panjang = System.in.read(data);
18             //Sistem.in.read mengembalikan panjang karakter yang
19             //diinputkan (termasuk enter yang dianggap 2 karakter..)
20             System.out.print("Yang anda ketik : ");
21             System.out.write(data);
22             System.out.println("Panjang Karakter : " + panjang);
23             System.out.print("index ke-1 sebnyk 3 : ");
24             System.out.write(data, 1, 3);
25         } catch (IOException e) {
26             System.out.print("Terjadi Exception");
27         }
28     }
29 }
```

## Output:

```
a[1] --- exec:3.1.0:exec (default-cli) @ 2211102160_Modul8 ---
Masukkan data : Fatimm
Yang anda ketik : Fatimm
Panjang Karakter : 7
index ke-1 sebnk 3 : ati
-----
BUILD SUCCESS
-----
Total time: 10.577 s
Finished at: 2024-05-26T15:01:19+07:00
-----
```

## Penjelasan:

Program `DemoStream4` dalam paket `GUIDED4` menggunakan aliran (stream) untuk membaca input dari pengguna dan menampilkannya kembali menggunakan metode `System.out.write`. Program ini juga menghitung panjang input yang dimasukkan oleh pengguna.

Proses dimulai dengan mendeklarasikan array byte `data` dengan panjang 10 untuk menyimpan input dari pengguna. Variabel `panjang` digunakan untuk menyimpan panjang input yang dibaca. Setelah meminta pengguna untuk memasukkan data melalui prompt "Masukkan data :", program menggunakan `System.in.read(data)` untuk membaca input pengguna dan menyimpannya ke dalam array `data`, serta mengembalikan panjang input yang dibaca.

Kemudian, program menampilkan pesan "Yang anda ketik :" sebelum mencetak input yang telah dibaca menggunakan `System.out.write(data)`. Selanjutnya, program mencetak panjang input yang dibaca, termasuk karakter newline yang dianggap sebagai dua karakter, dengan pesan "Panjang Karakter :". Terakhir, program mencetak tiga karakter mulai dari indeks ke-1 dari array `data` dengan pesan "index ke-1 sebnk 3 :".

## Output Program:

Jika pengguna memasukkan "Fatimm" diikuti dengan menekan Enter, outputnya akan menjadi:

```
```
Masukkan data : Fatimm
Yang anda ketik : Fatimm
Panjang Karakter : 7
index ke-1 sebnk 3 : atm
```
```

1. Masukkan data : :
  - Program meminta pengguna untuk memasukkan data.
2. Input Pengguna:
  - Pengguna mengetik "Fatimm" dan menekan Enter.
3. Yang anda ketik :
  - Program menampilkan pesan ini sebelum mencetak input pengguna.
4. Fatimm:
  - Program mencetak karakter-karakter yang telah dibaca dari input pengguna. Karena `System.out.write(data)` mencetak hingga panjang array

``data``, karakter yang diinputkan akan tercetak sampai batas panjang array atau sampai karakter valid terakhir.

6. Panjang Karakter : 7:

- Program mencetak panjang input yang dibaca. Panjang input sebenarnya adalah 6 karakter ("Fatimm") ditambah 1 karakter newline, total menjadi 7.

7. index ke-1 sebanyak 3 :

- Program menampilkan pesan ini sebelum mencetak tiga karakter mulai dari indeks ke-1.

8. `ati`:

- Program mencetak tiga karakter mulai dari indeks ke-1 dalam array ``data``, yaitu `"ati"` dari `"Fatimm"`.

Output tersebut menunjukkan bahwa program berhasil membaca input "Fatimm" yang dimasukkan oleh pengguna dan mencetaknya kembali. Setelah itu, program menampilkan panjang input, serta tiga karakter mulai dari indeks ke-1 dari input tersebut.