

Nama : Ilham Adi Muslim  
NIM : 20230040156  
Kelas : TI23F

## Percobaan 1

```
public class exception {  
    Run | Debug  
    public static void main(String[] args) {  
        int a[]=new int[5];  
        a[5]=100;  
    }  
}
```

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"  
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin exception "  
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Array index out of range: 5
```

Pendeklarasian a = 5 berarti hanya indeks 0 – 4 sedangkan a[5] mencoba mengakses array ke 5 maka terjadilah eror

```
public class exception {  
    Run | Debug  
    public static void main(String[] args) {  
        int a[]=new int[5];  
        a[5]=100;  
        try{  
            a[5]=100;  
        }  
        catch(Exception e){  
            System.out.println("Terjadi pelanggaran memory");  
        }  
    }  
}
```

```
C:\Us Focus folder in explorer (ctrl + click) \oding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"  
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin exception "  
Terjadi pelanggaran memory
```

Kode di perbaiki agar si eror dapat diberitahukan kepada user bahwa inputan melebihi kapasitas

## Percobaan 2

```

public class exception2 {
    Run | Debug
    public static void main(String[] args) {
        int i=0;
        String greeting[]={
            "Hello World!",
            "No, I mean it!",
            "Hello World"
        };
        while(i<4){
            System.out.println(greeting[i]);
            i++;
        }
    }
}

```

```

C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin exception2 "
Hello World!
No, I mean it!
Hello World
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Array index out of range: 3
    at exception2.main(exception2.java:10)

```

Greeting hanya memiliki 3 elemen yaitu indeks 0, 1, dan 2. Tetapi while ( $i < 4$ ) akan mencoba akses sampai greeting indeks ke 3, yang tidak ada. Maka pada  $i = 3$ , program akan mengalami error

```

public class exception2 {
    Run | Debug
    public static void main(String[] args) {
        int i=0;
        String greeting[]={
            "Hello World!",
            "No, I mean it!",
            "Hello World"
        };
        while(i<4){
            try{
                String[] greeting - exception2.main(String[])
                System.out.println(greeting[i]);
                i++;
            }
            catch(ArrayIndexOutOfBoundsException e){
                System.out.println("Resetting index value");
                i=0;
            }
        }
    }
}

```

```

No, I mean it!
Hello World
Resetting index value
Hello World!
No, I mean it!
Hello World
Resetting index value
Hello World!
No, I mean it!
Hello World
Resetting index value
Hello World!
No, I mean it!
Hello World
Resetting index value
Hello World!
No, I mean it!
Hello World
Resetting index value
Hello World!
No, I mean it!
Hello World

```

Blok try-catch digunakan untuk menangkap kesalahan `ArrayIndexOutOfBoundsException`. Saat  $i = 3$  dan error terjadi, program tidak berhenti. Sebaliknya Pesan "Resetting index value" ditampilkan dan Nilai  $i$  di-reset menjadi 0, sehingga loop diulang dari awal hal ini menimbulkan adanya loop tak berujung.

### Percobaan 3

```
src > J Exception3.java > Exception3 > main(String[])
1 public class Exception3 {
    Run | Debug
2     public static void main(String[] args) {
3         int bil=10;
4         System.out.println(bil/0);
5     }
6 }
```

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin Exception3 "
Exception in thread "main" java.lang.ArithmeticException: divide by zero
at Exception3.main(Exception3.java:4)
```

Program langsung crash dan berhenti karena program mencoba membagi 10 / 0. Ini menyebabkan ArithmeticException karena pembagian dengan nol tidak diperbolehkan dalam Java.

```
public class Exception3 {
    Run | Debug
    public static void main(String[] args) {
        int bil=10;
        try{
            System.out.println(bil/0);
        }
        catch(Exception e){
            System.out.println("Ini menghandle error yang terjadi");
        }
    }
}
```

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin Exception3 "
Ini menghandle error yang terjadi
```

pada code perbaikan pembagian bil / 0 tetap menimbulkan error tetapi karena diletakkan di dalam try-catch, error tersebut ditangkap (handled) oleh catch.

Maka, program tidak crash, dan akan mencetak:

*Ini menghandle error yang terjadi*

## Percobaan 4

```
public class CobaException4 {
    Run | Debug
    public static void main(String[] args) {
        int bil=10;
        String b[]={"a","b","c"};
        try{
            System.out.println(b[3]);
            System.out.println(bil/0);
        }
        catch(ArithmeticException e){
            System.out.println("Terjadi Aritmatika error");
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Melebihi jumlah array");
        }
        catch(Exception e){
            System.out.println("Ini menghandle error yang terjadi");
        }
    }
}
```

C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe" -cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin CobaException4 "

Melebihi jumlah array

b[3] menyebabkan `ArrayIndexOutOfBoundsException` karena hanya ada indeks 0–2 dan Java langsung menghentikan eksekusi dalam blok try setelah exception muncul. Maka baris `bil/0` tidak pernah dijalankan. Lalu error ditangkap oleh `catch(ArrayIndexOutOfBoundsException e)`, lalu mencetak:

*Melebihi jumlah array*

```
public class CobaException4 {
    Run | Debug
    public static void main(String[] args) {
        int bil=10;
        String b[]={"a","b","c"};
        try{
            System.out.println(bil/0);
            System.out.println(b[3]);
        }
        catch(ArithmeticException e){
            System.out.println("Terjadi Aritmatika error");
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Melebihi jumlah array");
        }
        catch(Exception e){
            System.out.println("Ini menghandle error yang terjadi");
        }
    }
}
```

C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe" -cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin CobaException4 "

Terjadi Aritmatika error

## Percobaan 5

Elemen	Fungsi
try	Menyimpan kode yang berpotensi menimbulkan error
catch (ArithmeticException)	Menangani kesalahan pembagian dengan nol (bil/0) secara spesifik
e.getMessage()	Memberi pesan singkat penyebab error ("/ by zero")
e.printStackTrace()	Menampilkan rincian letak error dan urutan pemanggilan method
catch (Exception e)	Penanganan umum jika ada error lain yang tidak tertangani sebelumnya

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin Exception5 "
Pesan error:
divide by zero
Info stack erase
java.lang.ArithmeticException: divide by zero
    at Exception5.main(Exception5.java:5)
java.lang.ArithmeticException: divide by zero
    at Exception5.main(Exception5.java:5)
```

## Percobaan 6

Komponen	Fungsi
throw t	Melempar exception NullPointerException secara eksplisit
System.out.println(...) setelah throw	Tidak dijalankan karena sudah keluar dari method
try	Menangkap error dari pemanggilan demo()
catch(NullPointerException)	Menangani error spesifik dan mencetak pesan dari exception
System.out.println("Selesai")	Tidak dijalankan karena eksekusi langsung lompat ke catch

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin ThrowExample "
Ada pesan error: java.lang.NullPointerException: Coba Throw
```

## Percobaan 7

Elemen Program	Fungsi
throw new Exception(...)	Melempar objek exception secara manual dengan pesan khusus
try-catch	Menangkap dan mengelola exception agar program tidak crash
getMessage()	Mengambil hanya isi pesan dari exception
toString()	Menampilkan tipe exception + pesan
printStackTrace()	Menampilkan info lengkap lokasi error terjadi

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin ThrowExample2 "
Caught Exception
e.getMessage():Here's my Exception
e.toString():java.lang.Exception: Here's my Exception
e.printStackTrace():
java.lang.Exception: Here's my Exception
    at ThrowExample2.main(ThrowExample2.java:4)
```

## Percobaan 8

```
import java.io.*;

public class Test3 {
    public void methodA(){
        System.out.println("Method A");
    }
    public void methodB() throws IOException{
        System.out.println(20/0);
        System.out.println("Method B");
    }
}

class Utama
{
    Run | Debug
    public static void main(String[] args) throws IOException{
        Test3 c=new Test3();
        c.methodA();
        c.methodB();
    }
}
```

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin Utama "
Method A
Exception in thread "main" java.lang.ArithmeticException: divide by zero
    at Test3.methodB(Test3.java:8)
    at Utama.main(Test3.java:17)
```

methodB() menyatakan throws IOException, namun tidak melempar IOException. Yang dilempar sebenarnya adalah **ArithmeticException** karena pembagian dengan nol (20/0).

ArithmeticException adalah **unchecked exception**, sehingga:

- Tidak perlu dideklarasikan dengan throws.
- Bisa langsung ditangani dengan try-catch jika diinginkan.

- Karena `main()` menyatakan `throws IOException`, maka compiler mengizinkan pemanggilan `methodB()` tanpa `try-catch`.
- Namun saat program dijalankan, akan muncul runtime error:

Exception in thread "main" java.lang.ArithmeticException: / by zero

- `IOException` **tidak dilempar** dalam program ini, sehingga `throws IOException` tidak berguna secara fungsional.

```
import java.io.*;

public class Test3 {
    public void methodA(){
        System.out.println("Method A");
    }
    public void methodB() throws IOException{
        System.out.println(20/0);
        System.out.println("Method B");
    }
}

class Utama{
    Run | Debug
    public static void main(String[] args){
        Test3 o=new Test3();
        o.methodA();
        try{
            o.methodB();
        }
        catch(Exception e){
            System.out.println("Error di Method B");
        }
        finally{
            System.out.println("Ini selalu dicetak");
        }
    }
}
```

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin Utama "
Method A
Error di Method B
Ini selalu dicetak
```

- `methodB()` tetap bisa dipanggil karena **ditangani oleh blok `try-catch`**.
- Saat `methodB()` dijalankan, `20 / 0` menyebabkan **`ArithmeticException`**, yang ditangkap oleh blok `catch(Exception e)`.
- Program tidak crash, dan akan mencetak:

```
Method A
Error di Method B
Ini selalu dicetak
```

## Percobaan 9

```
class Propagate {
    Run | Debug
    public static void main(String[] args){
        try{
            System.out.println(reverse(s:"This is a string"));
        }
        catch(Exception e){
            System.out.println("The String was blank");
        }
        finally{
            System.out.println("All done");
        }
    }

    public static String reverse(String s) throws Exception{
        if(s.length()==0){
            throw new Exception();
        }
        String reverseStr = "";
        for(int i=s.length()-1 ; i>=0 ; --i){
            reverseStr+=s.charAt(i);
        }
        return reverseStr;
    }
}
```

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin Propagate "
gnirts a si sihT
All done
```

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin Propagate "
The String was blank
All done
```

- Program awal mencetak string yang dibalik dan selalu mencetak "All done" di akhir karena `finally` selalu dijalankan.
- Jika method `reverse("This is a string")` dihapus:
  - Tidak ada output string terbalik.
  - Blok `catch` tidak dijalankan, karena tidak ada error.
  - Hanya "All done" yang dicetak dari blok `finally`.



## Percobaan 10

```
import java.io.IOException;
import java.io.RandomAccessFile;

public class Exception10 {
}

class RandomAccessRevisi{
    Run | Debug
    public static void main(String[] args) {
        String bookList[]={"Satu","Dua","Tiga"};
        int yearList[]={1920,1230,1940};
        try{
            RandomAccessFile books = new RandomAccessFile("books.txt","rw");
            for(int i=0;i<3;i++){
                books.writeUTF(bookList[i]);
                books.writeInt(yearList[i]);
            }
            books.seek(0);
            System.out.println(books.readUTF()+" "+books.readInt());
            System.out.println(books.readUTF()+" "+books.readInt());
            books.close();
        }
        catch(IOException e){
            System.out.println("Indeks melebihi batas");
        }
        System.out.println("test");
    }
}
```

- Program ini menulis dan membaca data dari file menggunakan `RandomAccessFile`.
- File `books.txt` akan berisi pasangan *String* dan *Integer* (judul buku dan tahun).
- Setelah menulis, program membaca kembali **dua data pertama** dan menampilkannya.

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin RandomAccessRevisi "
Satu 1920
Dua 1230
test
```

## Percobaan 11

```
C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin RangeErrorException "
Range error: Position 1
This is the last program.
```

```

class RangeErrorException extends Throwable{
    public RangeErrorException(String s){
        super(s);
    }
}

Run | Debug
public static void main(String[] args){
    int position=1;
    try{
        if(position>0){
            throw new RangeErrorException("Position " +position);
        }
    }
    catch(RangeErrorException e){
        System.out.println("Range error: " +e.getMessage());
    }
    System.out.println("This is the last program.");
}

```

Program ini mendemonstrasikan penggunaan exception kustom untuk memeriksa nilai variabel (`position`) dan menangani kesalahan dengan menampilkan pesan. Eksekusi tetap berlanjut setelah exception ditangani.

- `RangeErrorException` merupakan **custom exception** yang diturunkan langsung dari `Throwable`.
  - Biasanya lebih umum menurunkan dari `Exception`, bukan `Throwable`.
- Pada `main`, `position = 1`, dan kondisi `position > 0` terpenuhi → maka **exception dilempar**.
- Exception ditangkap, lalu menampilkan pesan:  
→ "Range error: Position 1"
- Program tetap berlanjut dan mencetak:  
→ "This is the last program."

## Percobaan 12

```

C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception> cmd /C ""C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe"
-cp C:\Users\ilham\MATERI\Semester4\PBO\Coding\Exception\exception\bin Eksepsi "
Tampil
OK!
Tampil
Exception generated by: amir
Tangkap:MyException

```

```

class MyException extends Exception{

    private String Teks;
    MyException(String s){
        Teks="Exception generated by: "+s;
        System.out.println(Teks);
    }
}

class Eksepsi{
    static void tampil(String s)throws Exception{
        System.out.println("Tampil");
        if(s.equals("amir")){
            throw new MyException(s);
        }
        System.out.println("OK!");
    }
}

Run | Debug
public static void main(String[] args)throws Exception{
    try{
        tampil(s:"ali");
        tampil(s:"amir");
    }
    catch(MyException ex){
        System.out.println("Tangkap:"+ex);
    }
}

```

Program ini mendemonstrasikan cara membuat dan menggunakan custom exception (MyException), serta menangkapnya secara eksplisit di blok try-catch. Exception hanya dilempar saat kondisi spesifik terpenuhi (s.equals("amir")), dan program tetap berjalan normal sebelum error terjadi.

- tampil("ali"):
  - Mencetak Tampil
  - Tidak memenuhi syarat s.equals("amir")
  - Mencetak OK!
- tampil("amir"):
  - Mencetak Tampil
  - Kondisi terpenuhi → throw new MyException("amir")
  - Constructor MyException mencetak: Exception generated by: amir
  - Exception ditangkap di catch → mencetak: Tangkap:MyException