

Subject :

- Method
- Exception

Objective :

- o Memahami Method
- o Memahami Exception

Tugas praktikum :**Praktikum Soal 1**

1. Buatlah file WhiteSpaceException.java.
2. Edit file WhiteSpaceException.java dengan sintaks seperti ini :

```
public class WhiteSpaceException extends Exception{
    public WhiteSpaceException() {
        super("Terjadi kesalahan : 'tidak boleh ada spasi
/ whitespace'");
    }

    public WhiteSpaceException(String text) {
        super("Terjadi kesalahan : 'tidak boleh ada spasi
/ whitespace', text : " + text);
    }
}
```

3. Compile file WhiteSpaceException.java menggunakan java compiler (javac) :

```
$ javac WhiteSpaceException.java
```

4. Setelah kompilasi berhasil, Buatlah file WhiteSpaceMain.java.
5. Edit file WhiteSpaceMain.java dengan sintaks seperti ini :

```
import java.util.Arrays;
import java.util.logging.Level;
import java.util.logging.Logger;

public class WhiteSpaceMain {
    public static void main(String[] args) {
        try {
            input();
        } catch (WhiteSpaceException ex) {
            Logger.getLogger(WhiteSpaceMain.class.
getName()).log(Level.SEVERE, null,
ex);
        }
    }
}
```

```

    }

    public static void input() throws
    WhiteSpaceException{
        String username = "TidakBolehMemakaiSpasi";

        char[] charArray = username.toCharArray();
        for (char c : charArray) {
            if(c == ' '){
                throw new WhiteSpaceException
                (username);
            }
        }

        System.out.println("Kalimat ini " +
        username);
    }
}

```

6. Compile file WhiteSpaceMain.java menggunakan java compiler (javac) :

```
$ javac WhiteSpaceMain.java
```

7. Setelah kompilasi berhasil, running file WhiteSpaceMain.java menggunakan perintah seperti berikut :

```
$ java WhiteSpaceMain
```

8. Setelah proses running berhasil, lihat hasil running, jika berhasil akan muncul tampilan seperti di bawah ini :

```
Kalimat ini TidakBolehMemakaiSpasi
```

9. Jika pada file WhiteSpaceMain.java, kalimat “TidakBolehMemakaiSpasi” di ganti “Tidak Boleh Memakai Spasi”, lalu compile ulang file WhiteSpaceMain.java , selanjutnya running kembali file WhiteSpaceMain, setelah running berhasil, dia akan mengeluarkan Exception yang telah anda buat :

```

SEVERE: null
WhiteSpaceException: Terjadi kesalahan : 'tidak boleh ada
spasi / whitespace', text : Tidak Boleh Memakai Spasi
    at WhiteSpaceMain.input(WhiteSpaceMain.java:20)
    at WhiteSpaceMain.main(WhiteSpaceMain.java:8)

```

Praktikum Soal 2

1. Buatlah file MethodReturnValues.java.
2. Edit file MethodReturnValues.java dengan sintaks seperti ini :

```
public class MethodReturnValues {  
    public static void main(String[] args) {  
        myMethod();  
        int myInt = 0;  
        System.out.println(myInt);  
        myInt = myMethod();  
        System.out.println(myInt);  
    }  
  
    static int myMethod() {  
        System.out.println("This computer is lagging!");  
        return 4;  
    }  
}
```

3. Compile file MethodReturnValues.java menggunakan java compiler (javac) :

```
$ javac MethodReturnValues.java
```

4. Setelah kompilasi berhasil, running file MethodReturnValues.java menggunakan perintah seperti berikut :

```
$ java MethodReturnValues
```

5. Setelah proses running berhasil, lihat hasil running, jika berhasil akan muncul tampilan seperti di bawah ini :

```
This computer is lagging!  
0  
This computer is lagging!  
4
```

Praktikum Soal 3

1. Buatlah file Person.java.
2. Edit file Person.java dengan sintaks seperti ini :

```
public class Person {  
    public static void simpleMessage(String name) {  
        System.out.println("Hello " + name);  
    }  
}
```

3. Compile file Person.java menggunakan java compiler (javac) :

```
$ javac Person.java
```

4. Setelah kompilasi berhasil, Buatlah file Method.java.

5. Edit file Method.java dengan sintaks seperti ini :

```
import java.util.Scanner;

public class Method {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        Person personObject = new Person();

        System.out.println("Enter your name here : ");
        String name = input.nextLine();

        personObject.simpleMessage(name);
    }
}
```

6. Compile file Method.java menggunakan java compiler (javac) :

```
$ javac Method.java
```

7. Setelah kompilasi berhasil, running file Method.java menggunakan perintah seperti berikut :

```
$ java Method
```

8. Setelah proses running berhasil, lihat hasil running, jika berhasil akan muncul tampilan seperti di bawah ini :

```
Enter your name here :
```

9. Setelah muncul tampilan diatas, coba kamu isi dengan nama lengkap kamu → klik ENTER, jika berhasil akan muncul tampilan seperti di bawah ini :

```
Enter your name here :
Rudi Pradana
Hello Rudi Pradana
```

Praktikum Soal 4

1. Buatlah file ExceptionTest.java.

2. Edit file ExceptionTest.java dengan sintaks seperti ini :

```
import java.util.*;
```

```

public class ExceptionTest {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int x = 1;

        do {
            try {
                System.out.println("Enter first num : ");
                int n1 = input.nextInt();
                System.out.println("Enter second num : ");
                int n2 = input.nextInt();
                int sum = n1/n2;
                System.out.println(sum);
                x = 2;

            } catch (Exception e) {
                System.out.println("You cant do that");
            }
        } while (x == 1);
    }
}

```

3. Compile file ExceptionTest.java menggunakan java compiler (javac) :

```
$ javac ExceptionTest.java
```

4. Setelah kompilasi berhasil, running file ExceptionTest.java menggunakan perintah seperti berikut :

```
$ java ExceptionTest
```

5. Setelah proses running berhasil, lihat hasil running, jika berhasil akan muncul tampilan seperti di bawah ini :

```
Enter first num :
```

6. Setelah muncul tampilan diatas, coba anda isi dengan angka, misalkan : 40 → klik ENTER, jika berhasil akan muncul tampilan seperti di bawah ini :

```
Enter first num :
```

```
40
```

```
Enter second num :
```

7. Setelah muncul tampilan diatas, coba anda isi kembali dengan angka, misalkan : 10 → klik ENTER, jika berhasil akan muncul tampilan seperti di bawah ini :

```
Enter first num :
```

```
40
```

```
Enter second num :
```

```
10
4
```

8. Setelah muncul tampilan diatas, coba anda ulangi kembali dengan input perintah pertama dengan nilai : 40, dan untuk perintah kedua : 0, → klik ENTER, jika berhasil akan muncul tampilan seperti di bawah ini :

```
Enter first num :
40
Enter second num :
0
You cant do that
Enter first num :
```

9. Setelah muncul tampilan diatas, perhatikan kalimat yang saya garis bawahin, itu adalah Exception Handling yang telah anda buat. Jika anda isi perintah pertama dengan angka : 40 dan perintah kedua anda isi dengan huruf, misalnya : e, → klik ENTER, maka dia akan melakukan perulangan Exception yang telah anda buat, seperti di bawah ini :

```
Enter first num :
You cant do that
Enter first num :
You cant do that
Enter first num :
You cant do that
.....
```

Praktikum Soal 5

1. Buatlah file TryCatchException.java.
2. Edit file TryCatchException.java dengan sintaks seperti ini :

```
import java.io.*;

public class TryCatchException {
    static BufferedReader input = new BufferedReader(new
    InputStreamReader(System.in));
    public static void main(String[] args) {
        try {
            System.out.println("Choose one of the options,
            (1-3) " + "\n\t1.) Steak" + "\n\t2.) Lobster" +
            "\n\t3.) Cheeseburger");
            System.out.print("Masukan Pilihan Anda : ");
            String choose = input.readLine();
            if(choose.equals("1")){
                System.out.println("Anda memilih
                Steak :-)");
            } else if(choose.equals("2")){
```

```

        System.out.println("Anda memilih Lobster
        :-D");
    } else if(choose.equals("3")){
        System.out.println("Anda memilih
        Cheeseburger ;-)");
    } else {
        throw new Exception("pilihan anda tidak
        ada !");
    }
} catch (Exception e) {
    System.out.println(e.getMessage());
}
}
}

```

3. Compile file TryCatchException.java menggunakan java compiler (javac) :

```
$ javac TryCatchException.java
```

4. Setelah kompilasi berhasil, running file TryCatchException.java menggunakan perintah seperti berikut :

```
$ java TryCatchException
```

5. Setelah proses running berhasil, lihat hasil running, jika berhasil akan muncul tampilan seperti di bawah ini :

```

Choose one of the options, (1-3)
1.) Steak
2.) Lobster
3.) Cheeseburger
Masukan Pilihan Anda :

```

6. Setelah proses running tampil seperti di atas, coba isi dengan pilihan yang sudah di sediakan 1, 2, atau 3
jika sudah di pilih → klik ENTER, lalu dia akan memilih sesuai nilai yang sudah di tentukan, seperti tampilan di bawah ini :

```

Masukan Pilihan Anda : 1
Anda memilih Steak :-)

```

7. Setelah keluar seperti di atas, coba running ulang, selanjutnya isi dengan pilihan yang tidak di sediakan, misalkan : 4, nanti dia akan mengeluarkan Exception yang telah kita buat, seperti di bawah ini :

```

Masukan Pilihan Anda : 4
pilihan anda tidak ada !

```