

Modul 8

Exception Handling Dan Method

Exception Handling

Dalam bagian ini, kita akan mempelajari teknik yang dipakai dalam Java dalam menangani kondisi yang tidak bisa dalam menjalankan operasi normal dalam program. Teknik ini dinamakan exception handling.

- Event yang terjadi ketika suatu kesalahan terjadi saat program dijalankan.
- Sering digunakan untuk akses sumberdaya non memori seperti akses file dan koneksi database

Pengertian Exception

Exception adalah sebuah peristiwa yang menjalankan alur proses normal pada program. Peristiwa ini bisaanya berupa kesalahan (error) dari beberapa bentuk. Ini disebabkan program kita berakhir tidak normal.

Menangani Exception

Untuk menangani exception dalam Java, kita gunakan blok try-catch-finally. Apa yang kita lakukan dalam program kita adalah kita menempatkan pernyataan yang mungkin menghasilkan exception dalam blok ini. Bentuk umum dari blok try-catch-finally adalah,

```
Try
{
    //tuliskan pernyataan yang dapat mengakibatkan exception
    //dalam blok ini
}
catch( <exceptionType1><varName1> )
{
    //tuliskan aksi apa dari program Anda yang dijalankan jika ada
    //exception tipe tertentu terjadi
}
```

```
catch( <exceptionTypen><varNamen> )
{
    //tuliskan aksi apa dari program Anda yang dijalankan jika ada
    //exception tipe tertentu terjadi
}
finally
{
    //tambahkan kode terakhir di sini
}
```

Exception dilemparkan selama eksekusi dari blok try dapat ditangkap dan ditangani dalam blok catch. Kode dalam blok finally selalu di-eksekusi. Berikut ini adalah aspek kunci tentang sintak dari konstruksi try-catch-finally:

- Notasi blok bersifat perintah
- Setiap blok try, terdapat satu atau lebih blok catch, tetapi hanya satu blok finally.
- Blok catch dan blok finally harus selalu muncul dalam konjungsi dengan blok try, dan diata surutan
- Blok try harus diikuti oleh paling sedikit satu blok catch ATAU satu blok finally, atau keduanya.
- Setiap blok catch mendefinisikan sebuah penanganan exception. Header dari blok catch harus membawa satu argumen, dimana exception pada blok tersebut akan ditangani.

Contoh Exception

a) Arithmetic Exception

```
public class cobaException
{
    public static void main(String args[])
    {
        int bil = 10;
        System.out.println(bil/0);
    }
}
```

b) Struktur Blok Try Catch

```
try
{
    kode program yang mungkin menghasilkan exception
}
catch(Exception xx)
{
}
catch(Exception xx)
{
}
}
```

Method/Function

Konsep Dasar

- Fungsi adalah bagian dari kode program yang mempunyai tugas spesifik.
- Fungsi akan dipanggil ke program utama atau fungsi yang lain bila dibutuhkan.
- Sebuah fungsi bisaanya digunakan untuk menangani suatu proses tertentu yang ada di dalam sebuah program.

Bentuk penulisan fungsi/method

Kita bisa menuliskan fungsi atau method dengan beberapa cara, dan method juga mempunyai beberapa macam tipe yaitu:

- void
- string
- double
- integer

Dari beberapa tipe method diatas semua harus mempunya nilai kembalian kecuali method yang bertipe void, dalam praktikum ini kita hanya akan membahas method yang bertipe void dan untuk method yang lainnya akan anda pelajari di mata kuliah pemrograman berorientasi objek. Berikut ini cara untuk menuliskan fungsi/method:

1. Method yang tidak mempunyai parameter

```
Void hitung()
{
    // Statement atau perintah yang dikerjakan
}
```

2. method yang mempunyai parameter

```
Void hitung2(int a,int b) // Method yang mempunyai parameter
{
    // Statement atau perintah yang dikerjakan
}
```

2.3 Contoh Aplikasi

a. Program yang tidak menggunakan fungsi/method

```
public class faktorial
{
    public static void main(String args[]){
        int nilai=1;
        int data=4;
        if(data==0||data==1){
            nilai=1;
        }
        else{
            for(int i=1; i<=data; i++){
                nilai*=i;
            }
        }
        System.out.println(data+" ! = "+nilai);
    }
}
```

b. Program yang menggunakan fungsi/method

```
public class faktorial5
{
    public void test(int data){           // Fungsi atau method
        int nilai=1;
        if(data==0||data==1){
            nilai=1;
        }
        else{
            for(int i=1; i<=data; i++){
                nilai*=i;
            }
        }
        System.out.println(data+"! = "+nilai);
    }

    public static void main(String args[]){
        faktorial5 f5=new faktorial5();
        f5.test(4);
        f5.test(5);
    }
}
```

Dapat anda perhatikan bahwa program “a” yang tidak menggunakan fungsi, maka proses yang berjalan hanya sekali dan angka yang dihitung nilai faktorialnya cenderung statis. Berbeda dengan program “b” yang menggunakan fungsi, proses yang berjalan bisa lebih satu kali, yaitu kita bisa melakukan pemanggilan fungsi tersebut secara berulang dengan nilai angka yang parameter faktorialnya dapat diubah dengan lebih dinamis.

Jalankan Netbeans, buat projek dengan nama faktorial_non_fungsi, kemudian tambahkan kode program sebagai berikut :

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6
7  package faktorial_non_fungsi;
8
9  /**
10   *
11   * @author dani
12   */
13  public class Faktorial_non_fungsi {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          int nilai = 1;
21          int data = 4;
22          if((data == 0) || (data ==1))
23          {
24              nilai = 1;
25          }
26          else
27          {
28              for (int i = 1; i <= data; i++ )
29              {
30                  nilai *= i;
31              }
32          }
33          System.out.println(data + " ! = " + nilai);
34      }
35  }
36
```

Praktek 1 : screen capture pada e-learning!

jalankan programnya, sehingga tampilan seperti berikut :



```
ant -f "/Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 2020 - 2021/PEMROG
init:
Deleting: /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 2020 - 2021/PEMR
deps-jar:
Updating property file: /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 20
Compiling 1 source file to /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL
compile-single:
run-single:
4 ! = 24
BUILD SUCCESSFUL (total time: 0 seconds)
```

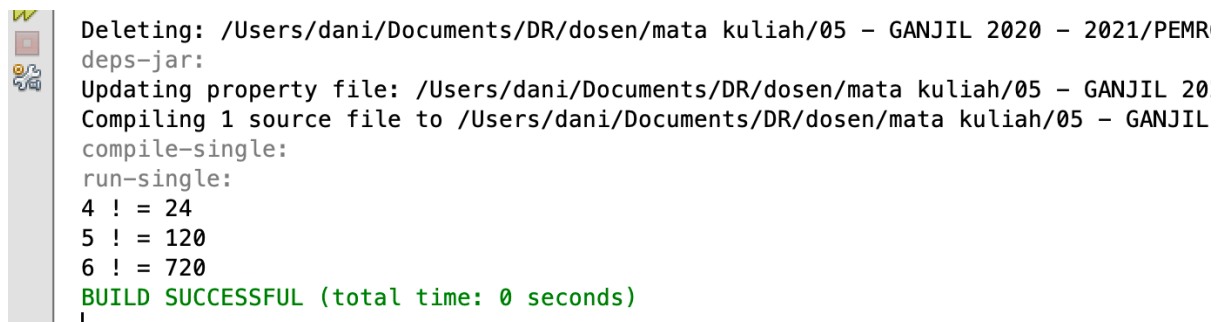
Praktek 2 : screen capture pada e-learning!

Jalankan Netbeans, buat proyek dengan nama faktorial_fungsi, kemudian tambahkan kode program sebagai berikut :

```
6
7 package faktorial_fungsi;
8
9 /**
10  *
11  * @author dani
12  */
13 public class Faktorial_fungsi {
14
15     /**
16     * @param args the command line arguments
17     */
18
19     public void eksekusi_faktorial(int data){
20
21         int nilai = 1;
22         if((data == 0) || (data ==1))
23         {
24             nilai = 1;
25         }
26         else
27         {
28             for (int i = 1; i <= data; i++ )
29             {
30                 nilai *= i;
31             }
32         }
33         System.out.println(data + " ! = " + nilai);
34     }
35
36     public static void main(String[] args) {
37         // TODO code application logic here
38         Faktorial_fungsi ff = new Faktorial_fungsi();
39         ff.eksekusi_faktorial(4);
40         ff.eksekusi_faktorial(5);
41         ff.eksekusi_faktorial(6);
42     }
43 }
```

Praktek 3 : screen capture pada e-learning!

jalankan programnya, sehingga tampilan seperti berikut :



```
Deleting: /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 2020 - 2021/PEMR
deps-jar:
Updating property file: /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 20
Compiling 1 source file to /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL
compile-single:
run-single:
4 ! = 24
5 ! = 120
6 ! = 720
BUILD SUCCESSFUL (total time: 0 seconds)
```

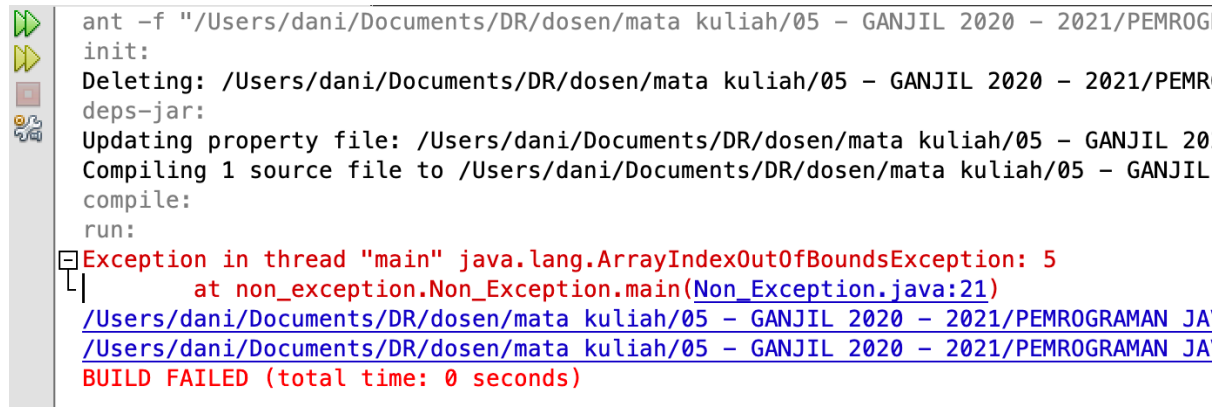
Praktek 4 : screen capture pada e-learning!

Jalankan Netbeans, buat projek dengan nama non_exception, kemudian tambahkan kode program sebagai berikut :

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  package non_exception;
8
9  /**
10   *
11   * @author dani
12   */
13  public class Non_Exception {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          int a[] = new int[5];
21          a[5]=100;
22      }
23
24  }
```

Praktek 5 : screen capture pada e-learning!

jalankan programnya, sehingga tampilan seperti berikut :



```
ant -f "/Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 2020 - 2021/PEMROGRAMAN JA
init:
Deleting: /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 2020 - 2021/PEMR
deps-jar:
Updating property file: /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 20
Compiling 1 source file to /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL
compile:
run:
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
    at non_exception.Non_Exception.main(Non_Exception.java:21)
BUILD FAILED (total time: 0 seconds)
```

Praktek 6 : screen capture pada e-learning!

Keterangan : program tidak akan sukses dijalankan, karena keterbatasan index array yang bisa dibuat!

Jalankan Netbeans, buat projek dengan nama exception, kemudian tambahkan kode program sebagai berikut :

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  package exception;
8
9  /**
10   *
11   * @author dani
12   */
13  public class Exception {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          int a[]=new int[5];
21          try
22          {
23              a[5]=100;
24          }
25          catch(java.lang.Exception e) // catch (Exception e)
26          {
27              System.out.println("Terjadi pelanggaran memory");
28          }
29      }
30
31  }
```

Praktek 7 : screen capture pada e-learning!

jalankan programnya, sehingga tampilan seperti berikut :



```
ant -f "/Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 2020 - 2021/PEMROG
init:
Deleting: /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 2020 - 2021/PEMR
deps-jar:
Updating property file: /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL 20
Compiling 1 source file to /Users/dani/Documents/DR/dosen/mata kuliah/05 - GANJIL
compile:
run:
Terjadi pelanggaran memory
BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktek 8 : screen capture pada e-learning!

Keterangan : Dengan penambahan exception, maka program bisa dijalankan, dan ketika program mengeksekusi perintah yang memungkinkan terjadinya kesalahan

dalam program, maka akan dilakukan exception dan selanjutnya menjalankan perintah yang kita kehendaki (misal menampilkan pesan kesalahan).