MINGGU II

Praktikum Pemrograman Berbasis Objek
oleh Asisten IF2210 2019/2020

OUTLINE

1. Assertion

ASSERTION

- Assertion adalah cara untuk memastikan program berjalan dengan baik
- Cukup tambahkan:
  ```
  assert x;
  ```
- x adalah boolean. x dapat diganti dengan ekspresi, misalnya:
  ```
  assert a == 0;
  ```

CONTOH ASSERTION

```
class Matrix {
  // ...
  public inverse() {
    Matrix result = new Matrix();
    // mengisi result
    // …
    // Memastikan hasil inverse bila dikalikan matrix ini
    // menghasilkan matriks identitas
    assert this.multiply(result).equals(IdentityMatrix());
    return result;
  }
}
```

CONTOH ASSERTION (2)

```
class A {
  // ...
  public calculate(int x) {
    if (x % 2 != 0) {
       // do something
    } else if (x % 3 != 0) {
       // do something
    } else {
       assert x % 6 == 0;
    }
  }
}
```

PENGGUNAAN ASSERTION

- Pada contoh sebelumnya, assertion digunakan untuk memverifikasi asumsi programmer
- Assertion tidak digunakan untuk memeriksa argumen, misalnya
    - Method inverse hanya dapat dilakukan pada matriks persegi
    - Gunakan Exception bila matriks bukan persegi
    - Supaya programmer lain dapat menangkap exception ini

MENGENABLE ASSERTION

- Defaultnya, JVM tidak akan menjalankan assertion yang ada di program
- Untuk meng-enable assertion, jalankan dengan flag -ea, misal:

```
javac A.java
java -ea A
```

BLACBOX TESTING

- Assertion juga digunakan pada blackbox testing
- Blackbox testing adalah pembuatan tes terhadap sebuah kode tanpa peduli bagaimana kode tersebut bekerja
- Dalam membuat testing, programmer harus teliti; kira-kira apa saja kode yang mungkin dibuat dan memiliki bug?
- Sebagai contoh, saat kalian mengumpulkan kode ke Olympia, kode kalian ditest dengan test yang dibuat oleh asisten. Artinya, asisten harus membuat berbagai macam testcase untuk memastikan kode kalian tidak memiliki bug.

CONTOH BLACKBOX TESTING

```
class Vector {
  ...
  public void size() { ... }
  public void push(int val) { ... }
  public void last() { ... }
}
```

Di sini kita mengasumsikan size() dan last() berjalan dengan baik dan mencoba mengetes apakah push() berjalan dengan baik

```
class VectorTest {
  public void test() {
    Vector v = new Vector();
    v.push(20);

    try {
      v.push(10);
    } catch (Exception e) {
      assert false; // seharusnya tidak ada exception
    }

    assert v.size() == 2;
    assert v.last() == 10;
  }
}
```

TAMBAHAN CATATAN

- Selain blackbox testing, ada juga whitebox testing. Test ini peduli mengenai bagaimana kode dibuat, misalnya pada soal ini:

Dari kedua kelas tersebut, Anda diminta **melanjutkan** sebuah *main program* yang telah dibuat.

```
// main.cpp
#include "A.hpp"
#include "B.hpp"

int main() {
  B b; /** ANDA HARUS MENGGUNAKAN DEFAULT CONSTRUCTOR */

  // Your code goes here :)
  // ...
  // ...

  b.print(); // OUTPUT: (888, 10888)
  return 0;
}
```

sehingga mampu menghasilkan output
(888, 10888)

Asisten memastikan Anda menggunakan polymorphism, bukan cara lain.

- Pada umumnya, testing java menggunakan framework seperti JUnit, jadi bukan menggunakan assert bawaan java