Unit-Testing using JUnit

Reported by:

Team	:	06
NIM/Name	:	1. 11321019/Elsaday Sianturi
		2. 11321044/Yudhi Purba
		3. 11321069/Maria Fransiska Giawa
		4. 11321071 / Putri Wita Marito

Testing for Unmodified Program

I. Testing for Palindrome_1.java

```
1.
2.
         Kelompok 06 PA2
3.
         1.
            11321019
                                Elsaday Sianturi
                                Yudhi Purba
4.
         2. 11321044
5.
         3.
            11321069
                                Maria Fransiska Giawa
6.
             11321071
                                Putri Wita Marito
7.
         * To change this license header, choose License Headers in Project Properties.
8.
9.
         * To change this template file, choose Tools | Templates
         * and open the template in the editor.
10.
11.
         */
         package JUnit_Palindrome1;
12.
13.
         import java.io.BufferedReader;
14.
         import\ java. io. Input Stream Reader;
15.
         /**
16.
         * @author USER
17.
18.
19.
         public class Palindrome1 {
20.
           public String methodPalindrome_1(int n1){
21.
             String hasil;
22.
             {\it BufferedReader} \ {\it InputStreamReader} ({\it System.in}));
23.
             int r, n2;
24.
             int rev=0;
25.
             n2=n1;
26.
             while(n1>0){
27.
               r = n1\%10;
               rev = rev*10+r;
28.
29.
               n1 = n1*10;
30.
31.
             if(rev==n2){}
               hasil = "palindrome number!";
32.
33.
34.
               hasil = "NOT palindrome number!";
35.
36.
             return hasil;
37.
38.
```

a. Input: 1

Snippet of test case

```
/**
  * Test of methodPalindrome_1 method, of class Palindrome1.
  */
@Test
public void testMethodPalindrome_1() {
    System.out.println("methodPalindrome_1");
    int n1 = 1;
    Palindrome1 instance = new Palindrome1();
    String expResult = "NOT palindrome number!";
    String result = instance.methodPalindrome_1(n1);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

b. Input: 22

Snippet of test case

```
public void testMethodPalindrome_1() {
    System.out.println("methodPalindrome_1");
    int n1 = 22;
    Palindrome1 instance = new Palindrome1();
    String expResult = "NOT palindrome number!";
    String result = instance.methodPalindrome_1(n1);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

```
JUnit_Palindrome1.JUnit_Palindrome1Test1 × JUnit_Palindrome1.JUnit_Palindrome1Test2 ×

Tests passed: 100.00 %

The test passed. (0.047 s)

methodPalindrome_1

methodPalindrome_1
```

c. Input: 27

Snippet of test case

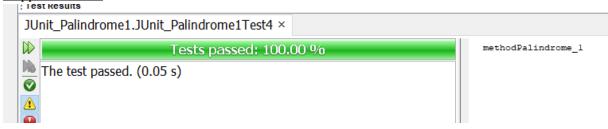
```
@Test
public void testMethodPalindrome_1() {
    System.out.println("methodPalindrome_1");
    int n1 = 27;
    Palindrome1 instance = new Palindrome1();
    String expResult = "NOT palindrome number!";
    String result = instance.methodPalindrome_1(n1);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

d. Input: 8998

Snippet of test case

```
@Test
public void testMethodPalindrome_1() {
    System.out.println("methodPalindrome_1");
    int n1 = 8998;
    Palindrome1 instance = new Palindrome1();
    String expResult = "NOT palindrome number!";
    String result = instance.methodPalindrome_1(n1);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results



e. Input: 2373 Snippet of test case

```
public void testMethodPalindrome_1() {
    System.out.println("methodPalindrome
    int n1 = 2373;
    Palindrome1 instance = new Palindrome1();
    String expResult = "NOT palindrome number!";
   String result = instance.methodPalindrome_1(n1);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
  // fail("The test case is a prototype.");
```

Snippet of results

JUnit_Palindrome1.JUnit_Palindrome1Test5 ×



f. Input: 78938 Snippet of test case

```
]
     public void testMethodPalindrome 1() {
         System.out.println("methodPalindrome_1");
         int n1 = 78938;
         Palindrome1 instance = new Palindrome1();
         String expResult = "NOT palindrome number!";
         String result = instance.methodPalindrome_1(n1);
         assertEquals(expResult, result);
         // TODO review the generated test code and remove the default call to fail.
       // fail("The test case is a prototype.");
```

Snippet of results

```
JUnit Palindrome1.JUnit Palindrome1Test6 ×
                                                                        methodPalindrome_1
                       Tests passed: 100.00 %
The test passed. (0.047 s)
O
```

g. Input: 1834554381 Snippet of test case

```
@Test
public void testMethodPalindrome_1() {
    System.out.println("methodPalindrome_1");
    int n1 = 1834554381;
    Palindrome1 instance = new Palindrome1();
    String expResult = "NOT palindrome number!";
    String result = instance.methodPalindrome_1(n1);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    // fail("The test case is a prototype.");
}
```

Snippet of results

h. Test Suite

Snippet of test case

```
package JUnit_Palindrome1;
14
15
      * @author USER
16
17
18 🗏 import org.junit.runner.RunWith;
    import org.junit.runners.Suite;
20
     @RunWith(Suite.class)
21
22
     @Suite.SuiteClasses({
23
         JUnit_Palindrome1Test1.class ,
         JUnit Palindrome1Test2.class ,
24
25
         JUnit_Palindrome1Test3.class ,
26
         JUnit_Palindrome1Test4.class ,
         JUnit_Palindrome1Test5.class ,
27
28
         JUnit_Palindrome1Test6.class ,
29
         JUnit_Palindrome1Test7.class ,
30
     public class JUnit_Palindrome2_TestSuite {
31
32
33
34
```

Snippet of results

```
Itest Results

JUnit_Palindrome1.JUnit_Palindrome2_TestSuite ×

Tests passed: 100.00 %

All 7 tests passed. (0.046 s)

All 7 tests passed. (0.046 s)

methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
```

II. Testing for Palindrome_2.java

```
1.
2.
         Kelompok 06 PA2
3.
         1.
             11321019
                                  Elsaday Sianturi
         2.
             11321044
                                  Yudhi Purba
4.
5.
         3.
              11321069
                                  Maria Fransiska Giawa
         4.
                                  Putri Wita Marito
6.
             11321071
7.
8.
          * To change this license header, choose License Headers in Project Properties.
9.
          * To change this template file, choose Tools | Templates
          ^{st} and open the template in the editor.
10.
11.
12.
         package [Unit_Palindrome2;
         import java.util.Scanner;
13.
14.
15.
          * @author USER
16.
17.
18.
         public class Palindrome2 {
19.
           public String methodPalindrome_2(String original){
20.
              String reverse = "";
21.
              String hasil;
22.
              Scanner in = new Scanner(System.in);
23.
24.
              int length = original.length();
25.
26.
             for(int i=length-1; i>=0; i--)
27.
                reverse = reverse + original.charAt(i);
28.
              if(original.equals(reverse))
29.
                hasil = "palindrome string!";
30.
              else
31.
                hasil = "NOT palindrome string!";
32.
              return hasil;
33.
           }
34.
```

a. Input: a

Snippet of test case

```
* Test of methodPalindrome_2 method, of class Palindrome2.
*/
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "is";
    Palindrome2 instance = new Palindrome2();
    String expResult = "NOT palindrome string!";
    String result = instance.methodPalindrome_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

b. Input: is

Snippet of test case

```
* Test of methodPalindrome_2 method, of class Palindrome2.
*/
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "is";
    Palindrome2 instance = new Palindrome2();
    String expResult = "NOT palindrome string!";
    String result = instance.methodPalindrome_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

```
JUnit_Palindrome2.JUnit_Palindrome2Test2 ×

Tests passed: 100.00 %

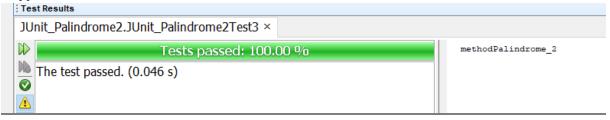
The test passed. (0.047 s)

The test passed. (0.047 s)
```

c. Input: isi

Snippet of test case

```
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "isi";
    Palindrome2 instance = new Palindrome2();
    String expResult = "palindrome string!";
    String result = instance.methodPalindrome_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```



d. Input: radar

Snippet of test case

```
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "radar";
    Palindrome2 instance = new Palindrome2();
    String expResult = "palindrome string!";
    String result = instance.methodPalindrome_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

```
JUnit_Palindrome2.JUnit_Palindrome2Test3 × JUnit_Palindrome2.JUnit_Palindrome2Test4 ×

Tests passed: 100.00 %

The test passed. (0.047 s)

The test passed. (0.047 s)
```

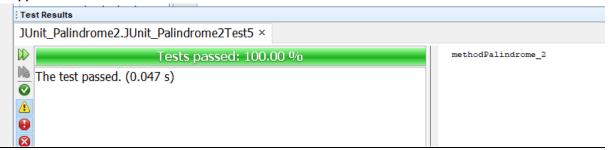
e. Input: palindrome

Snippet of test case

```
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "palindrome";
    Palindrome2 instance = new Palindrome2();
    String expResult = "NOT palindrome string!";
    String result = instance.methodPalindrome_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

2021/2022 | IT Del

Snippet of results



1132204 - PKPL

f. Input: nababan

Snippet of test case

```
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "nababan";
    Palindrome2 instance = new Palindrome2();
    String expResult = "palindrome string!";
    String result = instance.methodPalindrome_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

```
JUnit_Palindrome2.JUnit_Palindrome2Test5 × JUnit_Palindrome2.JUnit_Palindrome2Test6 ×

Tests passed: 100.00 %

The test passed. (0.049 s)

methodPalindrome_2
```

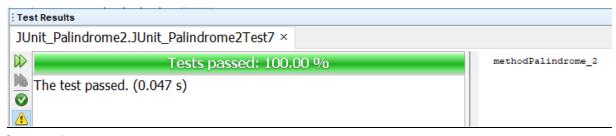
g. Input: read Snippet of test case

```
@Test

| public void testMethodPalindrome_2() {
| System.out.println("methodPalindrome_2");
| String original = "read";
| Palindrome2 instance = new Palindrome2();
| String expResult = "NOT palindrome string!";
| String result = instance.methodPalindrome_2(original);
| assertEquals(expResult, result);
| // TODO review the generated test code and remove the default call to fail.
| //fail("The test case is a prototype.");
| }
```

Snippet of results

2021/2022 | IT Del



h. TestSuite

Snippet of test case

```
package JUnit_Palindrome2;

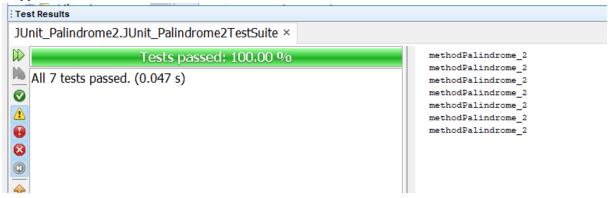
/**

* @author USER

*/
import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Suite.class)
@Suite.SuiteClasses({
    JUnit_Palindrome2Test1.class ,
    JUnit_Palindrome2Test2.class ,
    JUnit_Palindrome2Test3.class ,
    JUnit_Palindrome2Test4.class ,
    JUnit_Palindrome2Test5.class ,
    JUnit_Palindrome2Test5.class ,
    JUnit_Palindrome2Test6.class ,
    JUnit_Palindrome2Test7.class ,
})
public class JUnit_Palindrome2TestSuite {
```

Snippet of results



III. Testing for Reverse_1.java

```
/*
1.
         Kelompok 06 PA2
         1. 11321019
                                Elsaday Sianturi
2.
3.
         2.
             11321044
                                Yudhi Purba
            11321069
                                Maria Fransiska Giawa
4.
         3.
                                Putri Wita Marito
5.
            11321071
6.
         * To change this license header, choose License Headers in Project Properties.
7.
         * To change this template file, choose Tools | Templates
8.
         * and open the template in the editor.
9.
10.
         package JUnit_Reverse1;
11.
12.
```

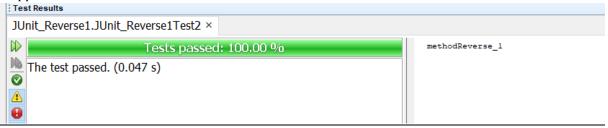
```
13.
         import java.io.BufferedReader;
14.
         import java.io.InputStreamReader;
15.
16.
         * @author USER
17.
18.
         */
19.
         public class Reverse1 {
20.
           public String methodReverse_1(int n){
21.
             String hasil;
22.
             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
23.
             int r;
24.
             int rev = 0;
25.
             int number = n;
26.
27.
             while(n>0){}
28.
               r = n\%10;
29.
               rev = rev*10+r;
30.
               n = n/10;
31.
             }
32.
             hasil = "The reverse of "+number+ " is "+rev;
33.
34.
             return hasil;
35.
           }
36.
```

a. Input: 1

Snippet of test case

```
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 1;
    Reversel instance = new Reversel();
    String expResult = "The reverse of "+n+ " is 1";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

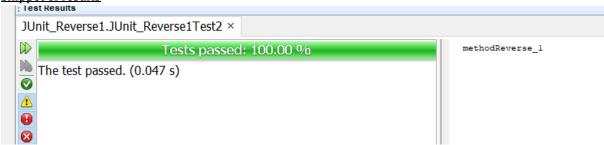


b. Input: 22

Snippet of test case

```
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 22;
    Reversel instance = new Reversel();
    String expResult = "The reverse of "+n+ " is 22";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results



c. Input: 27

Snippet of test case

```
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 27;
    Reversel instance = new Reversel();
    String expResult = "The reverse of "+n+ " is 72";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

d. Input: 8998 Snippet of test case

```
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 8998;
    Reverse1 instance = new Reverse1();
    String expResult = "The reverse of "+n+ " is 8998";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

e. Input: 2373

Snippet of test case

```
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 2373;
    Reversel instance = new Reversel();
    String expResult = "The reverse of "+n+" is 2373";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // Todo review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

```
Init_Reverse1.JUnit_Reverse1Test4 × JUnit_Reverse1Test5 ×

Tests passed: 0.00 %

No test passed, 1 test failed. (0.063 s)

JUnit_Reverse1.JUnit_Reverse1Test5 Failed

Junit_Reverse1.Junit_Reverse1Test5 Failed

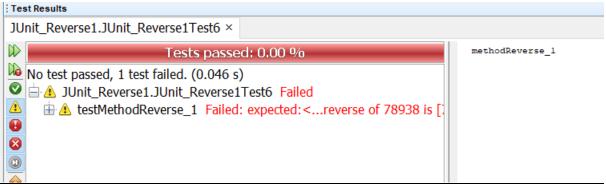
testMethodReverse_1 Failed: expected:<... reverse of 2373 is [2]
```

f. Input: 78938

Snippet of test case

```
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 78938;
    Reversel instance = new Reversel();
    String expResult = "The reverse of "+n+" is 78938";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    fail("The test case is a prototype.");
}
```

Snippet of results



g. Input: 1834554381

Snippet of test case

```
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 1834554381;
    Reversel instance = new Reversel();
    String expResult = "The reverse of "+n+" is 1834554381";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //ail("The test case is a prototype.");
}
```

```
JUnit_Reverse1.JUnit_Reverse1Test6 × JUnit_Reverse1.JUnit_Reverse1Test7 ×

Tests passed: 100.00 %

The test passed. (0.047 s)

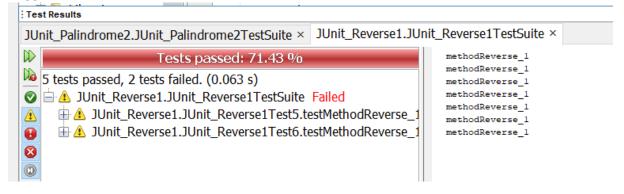
The test passed. (0.047 s)
```

h. TestSuite

Snippet of test case

```
package JUnit Reversel;
13
14
15
      * @author USER
16
17
18
  import org.junit.runner.RunWith;
   import org.junit.runners.Suite;
19
20
21
     @RunWith (Suite.class)
22
     @Suite.SuiteClasses({
23
         JUnit ReverselTest1.class ,
24
         JUnit ReverselTest2.class ,
25
         JUnit ReverselTest3.class ,
26
         JUnit ReverselTest4.class ,
27
         JUnit ReverselTest5.class ,
28
         JUnit ReverselTest6.class ,
29
         JUnit_ReverselTest7.class ,
30
     1)
31
     public class JUnit ReverselTestSuite {
32
33
34
```

Snippet of results



IV. Testing for Reverse_2.java

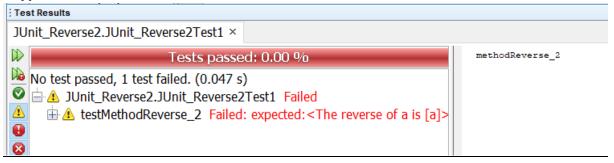
```
IV.
V.
         Kelompok 06 PA2
                                 Elsaday Sianturi
VI.
         1. 11321019
         2.
             11321044
                                 Yudhi Purba
VII.
VIII.
         3. 11321069
                                 Maria Fransiska Giawa
IX.
         4.
            11321071
                                 Putri Wita Marito
X.
XI.
          * To change this license header, choose License Headers in Project Properties.
          * To change this template file, choose Tools | Templates
XII.
XIII.
          * and open the template in the editor.
XIV.
         package JUnit_Reverse2;
XV.
XVI.
XVII.
XVIII.
XIX.
          * @author USER
XX.
          */
XXI.
         public class Reverse2 {
XXII.
            public String methodReverse_2(String original){
XXIII.
             String hasil;
             String reverse = "";
XXIV.
XXV.
             int length = original.length();
XXVI.
             for(int i=length-1; i>0; i--)
XXVII.
               reverse = reverse + original.charAt(i);
XXVIII.
XXIX.
             hasil = "The reverse of "+original+" is "+reverse;
XXX.
             return hasil;
XXXI.
           }
XXXII.
```

a. Input: a

Snippet of test case

```
/**
  * Test of methodReverse_2 method, of class Reverse2.
  */
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "a";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is a";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

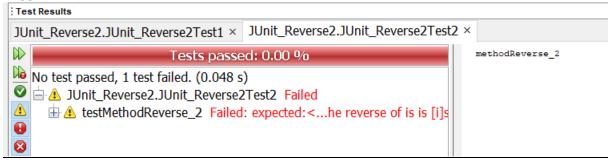


b. Input: is

Snippet of test case

```
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "is";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is is";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results



c. Input: isi

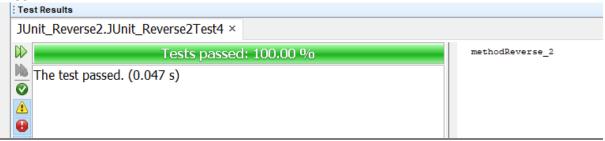
Snippet of test case

```
#/
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "isi";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is is";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

d. Input: radar Snippet of test case

```
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "radar";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is rada";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```



e. Input: palindrome

Snippet of test case

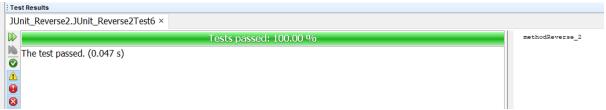
```
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "palindrome";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is emordi";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

f. Input: nababan Snippet of test case

```
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "nababan";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is nababa";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

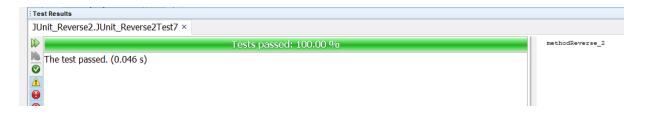


g. Input: read

Snippet of test case

```
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "read";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is dae";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

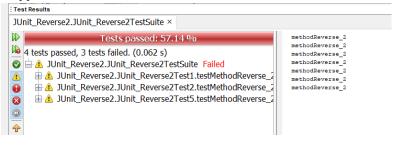


h. Test Suite

Snippet of test case

```
20
21
     @RunWith(Suite.class)
22
     @Suite.SuiteClasses({
23
        JUnit_Reverse2Test1.class ,
        JUnit_Reverse2Test2.class ,
24
        JUnit_Reverse2Test3.class ,
25
26
        JUnit_Reverse2Test4.class ,
27
        JUnit_Reverse2Test5.class ,
28
        JUnit_Reverse2Test6.class ,
29
        JUnit_Reverse2Test7.class ,
30
31
     public class JUnit_Reverse2TestSuite {
32
33
34
```

Snippet of results



Testing for Modified Program

I. Testing for Reverse1

```
1.
         Kelompok 06 PA2
2.
             11321019
                                 Elsadav Sianturi
3.
         1.
4.
             11321044
                                Yudhi Purba
                                 Maria Fransiska Giawa
5.
         3.
             11321069
                                Putri Wita Marito
6.
             11321071
7.
8.
         * To change this license header, choose License Headers in Project Properties.
         * To change this template file, choose Tools | Templates
9.
10.
         * and open the template in the editor.
11.
12.
         package JUnit_Reverse1;
13.
         import java.io.BufferedReader;
14.
15.
         import java.io.InputStreamReader;
16.
17.
18.
19.
         * @author USER
20.
21.
         public class Reverse1 {
           public String methodReverse_1(int n){
22.
23.
             String hasil;
24.
             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
25.
26.
             int rev = 0;
27.
             int number = n;
28.
29.
             while(n>0){}
30.
               r = n\%10;
               rev = rev*10+r;
31.
32.
               n = n/10;
             }
33.
34.
             hasil = "The reverse of "+number+ " is "+rev;
35.
36.
             return hasil;
37.
38.
```

a. Input: 1

Snippet of test case

```
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 1;
    Reverse1 instance = new Reverse1();
    String expResult = "The reverse of "+n+ " is 1";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

```
JUnit_Reverse1.JUnit_Reverse1Test2 ×

Tests passed: 100.00 %

The test passed. (0.047 s)

The test passed. (0.047 s)
```

b. Input: 22

Snippet of test case

```
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 22;
    Reverse1 instance = new Reverse1();
    String expResult = "The reverse of "+n+ " is 22";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

```
JUnit_Reverse1.JUnit_Reverse1Test2 ×

Tests passed: 100.00 %

The test passed. (0.047 s)

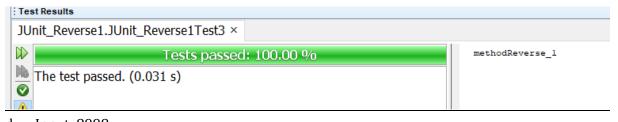
The test passed. (0.047 s)
```

c. Input: 27

Snippet of test case

```
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 27;
    Reverse1 instance = new Reverse1();
    String expResult = "The reverse of "+n+" is 72";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results



d. Input: 8998

```
Snippet of test case
```

```
Public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 8998;
    Reversel instance = new Reversel();
    String expResult = "The reverse of "+n+" is 8998";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

```
JUnit_Reverse1.JUnit_Reverse1Test4 ×

Tests passed: 100.00 %

The test passed. (0.063 s)

The test passed. (0.063 s)
```

e. Input: 2373

Snippet of test case

```
%/
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 2373;
    Reverse1 instance = new Reverse1();
    String expResult = "The reverse of "+n+ " is 2373";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

```
Test Results
JUnit_Reverse1.JUnit_Reverse1Test5 ×
                                        Tests passed: 100.00 %
The test passed. (0.047 s)
0
⚠
       Input: 78938
f.
Snippet of test case
      @Test
彐
      public void testMethodReverse 1() {
          System.out.println("methodReverse 1");
          int n = 78938;
          Reverse1 instance = new Reverse1();
          String expResult = "The reverse of "+n+ " is 78938";
          String result = instance.methodReverse_1(n);
          assertEquals(expResult, result);
          // TODO review the generated test code and remove the default call to fail.
          fail("The test case is a prototype.");
```

```
Unit_Reverse1.JUnit_Reverse1Test6 ×

Tests passed: 100.00 %

The test passed. (0.047 s)
```

g. Input: 1834554381

Snippet of test case

```
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 1834554381;
    Reversel instance = new Reversel();
    String expResult = "The reverse of "+n+" is 1834554381";
    String result = instance.methodReverse_1(n);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //ail("The test case is a prototype.");
}
```

2021/2022 | IT Del 1132204 - PKPL

Snippet of results

```
JUnit_Reverse1.JUnit_Reverse1Test6 × JUnit_Reverse1.JUnit_Reverse1Test7 ×

Tests passed: 100.00 %

The test passed. (0.047 s)

The test passed. (0.047 s)
```

h. TestSuite

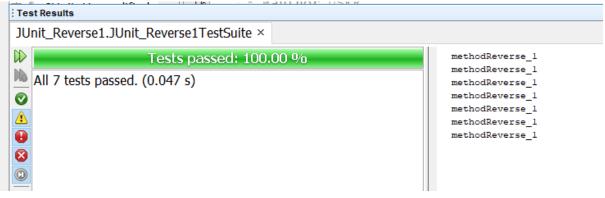
Snippet of test case

```
import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Suite.class)
@Suite.SuiteClasses({
    JUnit_ReverselTest1.class ,
    JUnit_ReverselTest2.class ,
    JUnit_ReverselTest3.class ,
    JUnit_ReverselTest4.class ,
    JUnit_ReverselTest5.class ,
    JUnit_ReverselTest5.class ,
    JUnit_ReverselTest6.class ,
    JUnit_ReverselTest7.class ,
})

public class JUnit_ReverselTestSuite {
```

Snippet of results



II. Testing for Reverse2

```
2.
3.
         Kelompok 06 PA2
                                 Elsaday Sianturi
4.
         1.
             11321019
            11321044
                                 Yudhi Purba
5.
         2.
6.
         3.
             11321069
                                 Maria Fransiska Giawa
7.
              11321071
                                 Putri Wita Marito
8.
9.
          * To change this license header, choose License Headers in Project Properties.
10.
          * To change this template file, choose Tools | Templates
          * and open the template in the editor.
11.
12.
13.
         package JUnit_Reverse2;
14.
15.
16.
          * @author USER
17.
18
         public class Reverse2 {
19.
20.
             public String methodReverse_2(String original){
             String hasil;
21.
22.
             String reverse = "";
             int length = original.length();
23.
24.
             for(int i=length-1; i>0; i--)
25.
               reverse = reverse + original.charAt(i);
26.
             hasil = "The reverse of "+original+" is "+reverse;
27.
28.
             return hasil;
29.
           }
30.
```

a. Input: a

Snippet of test case

```
/**
  * Test of methodReverse_2 method, of class Reverse2.
  */
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "a";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is a";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

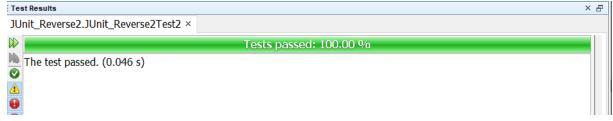
Snippet of results

```
| Itest Results | State | JUnit_Reverse1.JUnit_Reverse1.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reverse2.JUnit_Reve
```

b. Input: is

Snippet of test case

```
*/
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "is";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is is";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

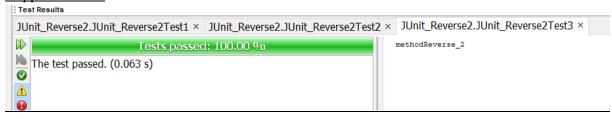


c. Input: isi

Snippet of test case

```
#/
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "isi";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is is";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results



d. Input: radar Snippet of test case

```
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "radar";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is rada";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

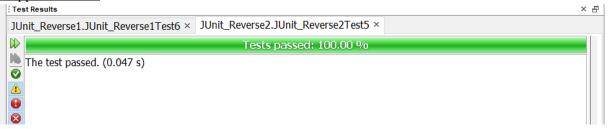


e. Input: palindrome

Snippet of test case

```
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "palindrome";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is emordi";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

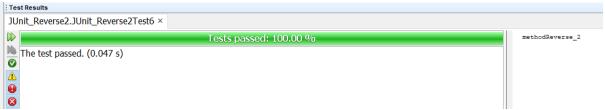


f. Input: nababan

Snippet of test case

```
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "nababan";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is nababa";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results



g. Input: read

Snippet of test case

```
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "read";
    Reverse2 instance = new Reverse2();
    String expResult = "The reverse of "+original+" is dae";
    String result = instance.methodReverse_2(original);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}
```

Snippet of results

2021/2022 | IT Del 1132204 - PKPL

h. TestSuite

Snippet of test case

```
package JUnit_Reverse2;

/**

* @author USER

*/

import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Suite.class)

@Suite.SuiteClasses({

    JUnit_Reverse2Test1.class ,
    JUnit_Reverse2Test2.class ,
    JUnit_Reverse2Test3.class ,
    JUnit_Reverse2Test4.class ,
    JUnit_Reverse2Test5.class ,
    JUnit_Reverse2Test5.class ,
    JUnit_Reverse2Test6.class ,
    JUnit_Reverse2Test7.class ,
    JUnit_Reverse2Test7.class ,
    Junit_Reverse2Test7.class ,
    Junit_Reverse2Test7.class ,
    Junit_Reverse2Test7.class ,
    Junit_Reverse2Test8uite {
```

Snippet of results

