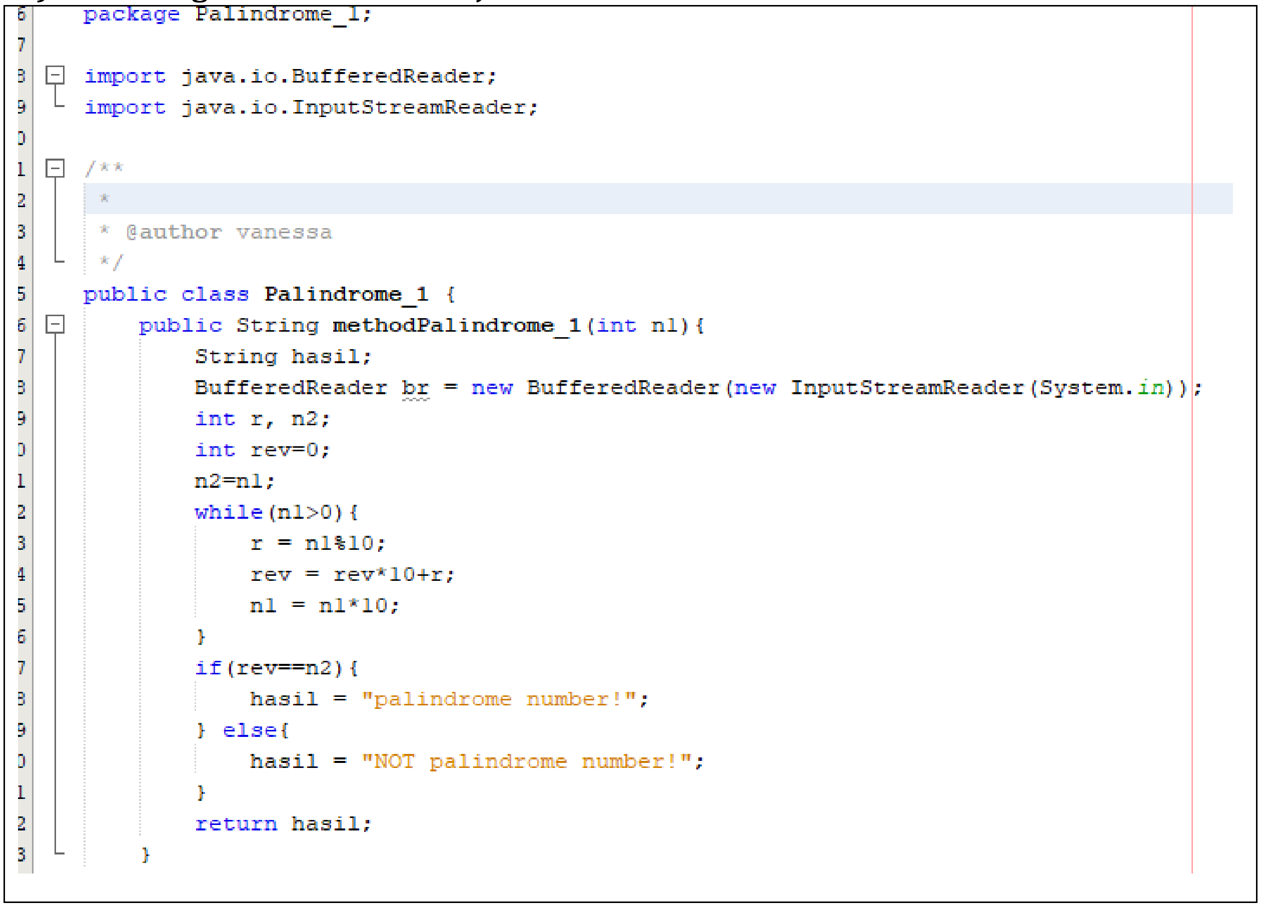
Unit-Testing using JUnit

Reported by:

Team	:	14
NIM/Name	:	1. Horas MP Saragih Sidabalok – 11322016 2. Mario Andreas Manurung -11322030 3. Nania Avantika Oligiviana Pangaribuan - 11322045 4. Vanessa Siahaan - 11322060

I. UNMODIFED

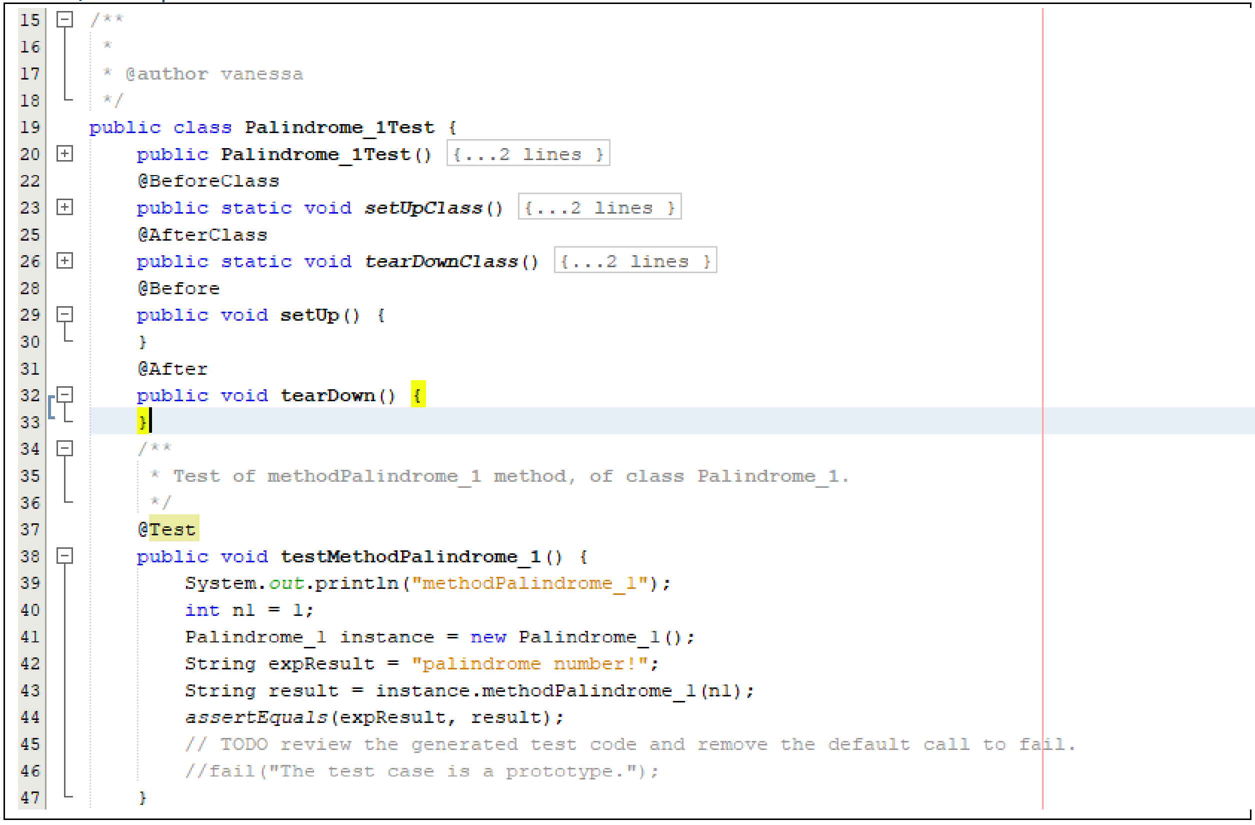
1) Testing for Palindrome_1.java



```
package Palindrome_1;

import java.io.BufferedReader;
import java.io.InputStreamReader;

/**
 *
 * @author vanessa
 */
public class Palindrome_1 {
    public String methodPalindrome_1(int n1){
        String hasil;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int r, n2;
        int rev=0;
        n2=n1;
        while(n1>0){
            r = n1%10;
            rev = rev*10+r;
            n1 = n1*10;
        }
        if(rev==n2){
            hasil = "palindrome number!";
        } else{
            hasil = "NOT palindrome number!";
        }
        return hasil;
    }
```

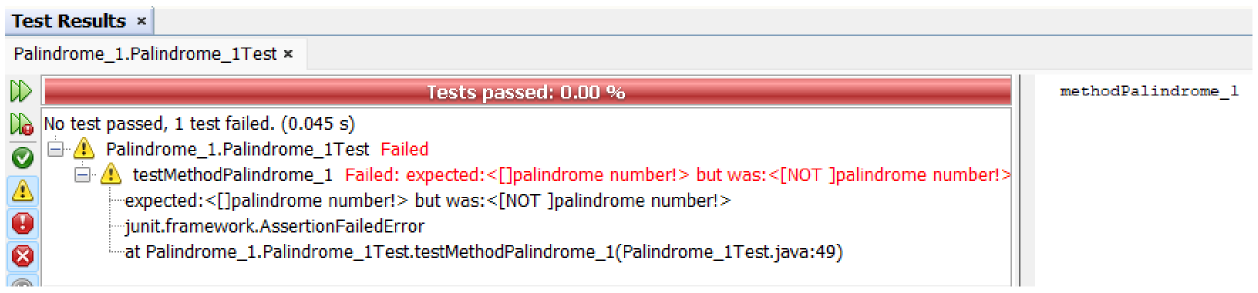
a. Input: 1

- **Snippet of test case**



```
15  /**
16   *
17   * @author vanessa
18   */
19  public class Palindrome_1Test {
20      public Palindrome_1Test() {...2 lines }
22      @BeforeClass
23      public static void setUpClass() {...2 lines }
25      @AfterClass
26      public static void tearDownClass() {...2 lines }
28      @Before
29      public void setUp() {
30      }
31      @After
32      public void tearDown() {
33      }
34      /**
35       * Test of methodPalindrome_1 method, of class Palindrome_1.
36       */
37      @Test
38      public void testMethodPalindrome_1() {
39          System.out.println("methodPalindrome_1");
40          int n1 = 1;
41          Palindrome_1 instance = new Palindrome_1();
42          String expResult = "palindrome number!";
43          String result = instance.methodPalindrome_1(n1);
44          assertEquals(expResult, result);
45          // TODO review the generated test code and remove the default call to fail.
46          //fail("The test case is a prototype.");
47      }
```

➢ **Snippet of results**



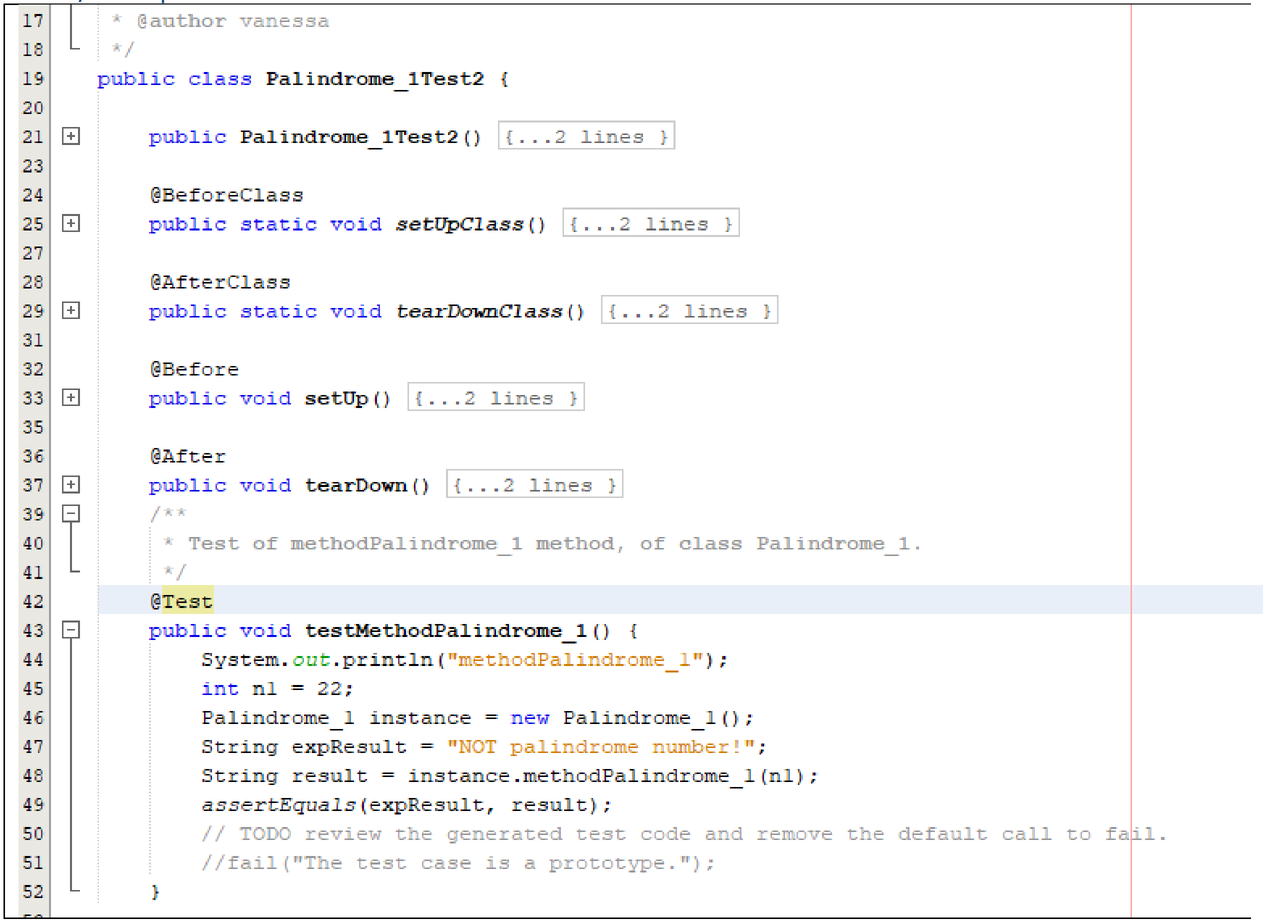
```
No test passed, 1 test failed. (0.045 s)
  Palindrome_1.Palindrome_1Test  Failed
    testMethodPalindrome_1  Failed: expected:<[]palindrome number!> but was:<[NOT ]palindrome number!>
      expected:<[]palindrome number!> but was:<[NOT ]palindrome number!>
      junit.framework.AssertionFailedError
      at Palindrome_1.Palindrome_1Test.testMethodPalindrome_1(Palindrome_1Test.java:49)
```

```
methodPalindrome_1
```

Penjelasan : Sebenarnya 1 merupakan bilangan palindrome dikarenakan jika dibalikkan tetap hasilnya sama namun mengapa terjadi error dikarenakan pada method menggunakan n1 = n1 * 10 seharusnya menggunakan n1 = n1 / 10 untuk menghindari perulangan tak terbatas dan memperbaiki perhitungan balik yang benar. Jika kita memeodif maka pada line 45 agar test case testMethodPalindrome_1 berhasil, expResult harus diubah menjadi "NOT palindrome number!" sesuai dengan output yang diharapkan dari fungsi methodPalindrome_1 untuk bilangan satu digit.

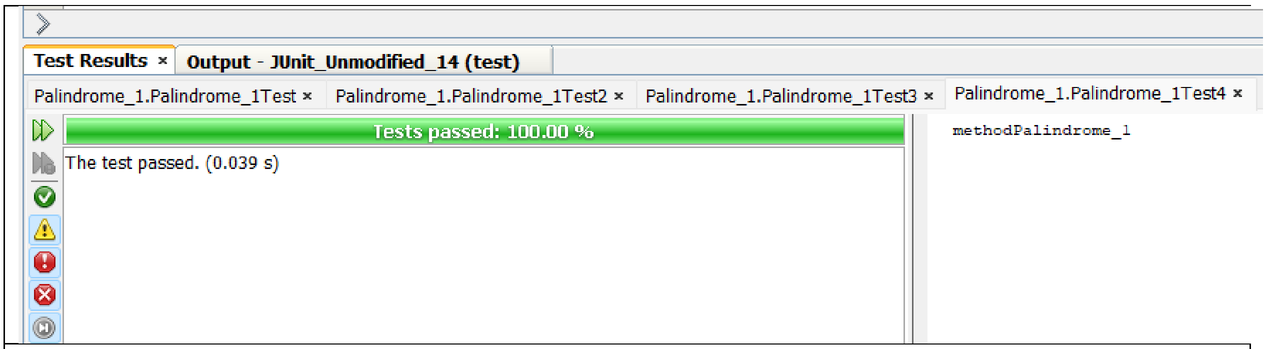
b. Input: 22

➢ **Snippet of test case**



```
 * @author vanessa
 */
public class Palindrome_1Test2 {

    public Palindrome_1Test2() {...2 lines }

    @BeforeClass
    public static void setUpClass() {...2 lines }

    @AfterClass
    public static void tearDownClass() {...2 lines }

    @Before
    public void setUp() {...2 lines }

    @After
    public void tearDown() {...2 lines }
    /**
     * Test of methodPalindrome_1 method, of class Palindrome_1.
     */
    @Test
    public void testMethodPalindrome_1() {
        System.out.println("methodPalindrome_1");
        int n1 = 22;
        Palindrome_1 instance = new Palindrome_1();
        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**



Test Results
Output - JUnit_Unmodified_14 (test)
Palindrome_1.Palindrome_1Test
Palindrome_1.Palindrome_1Test2
Palindrome_1.Palindrome_1Test3
Palindrome_1.Palindrome_1Test4

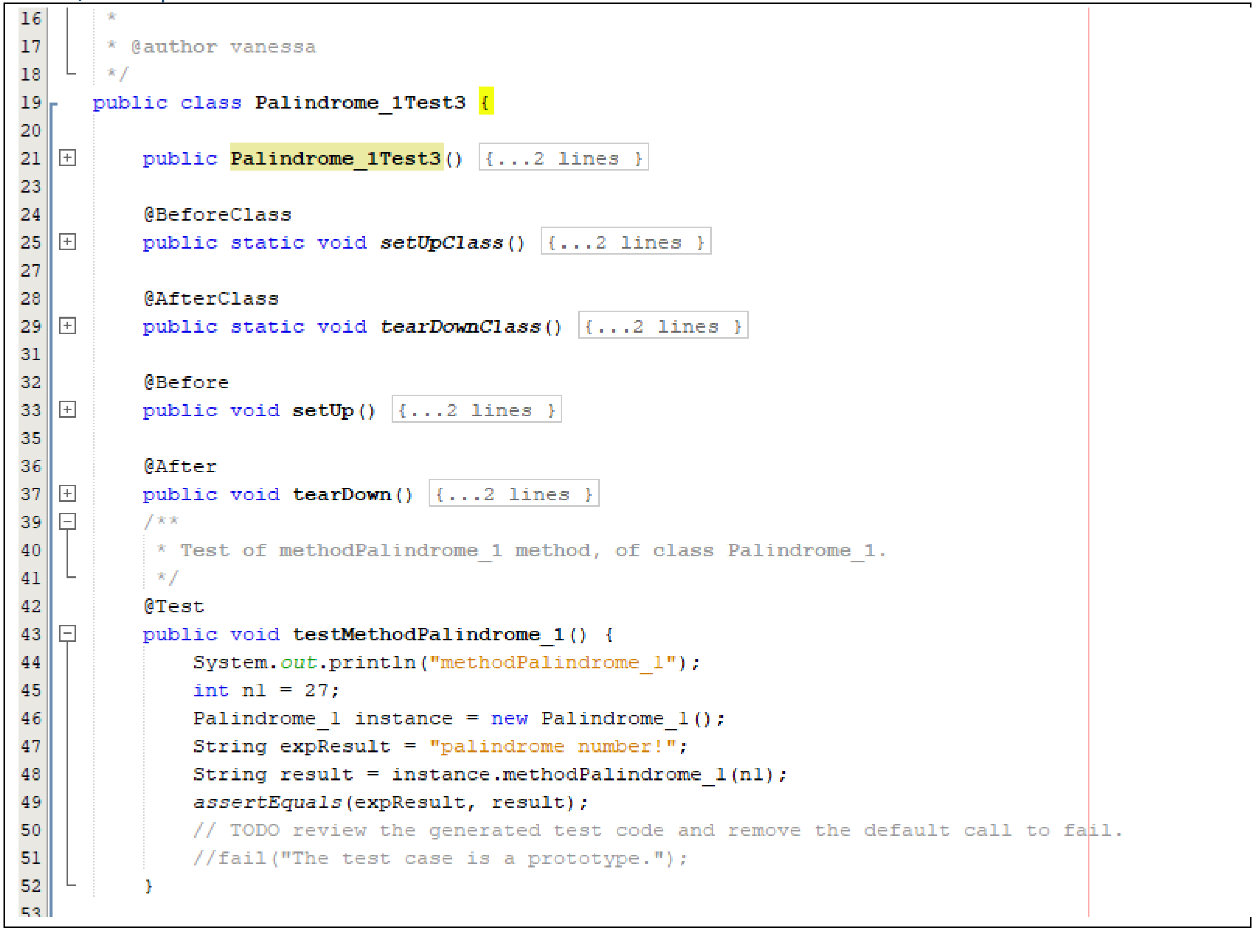
```
Tests passed: 100.00 %
The test passed. (0.039 s)
```

```
methodPalindrome_1
```

Penjelasan : Test berhasil bukan karena bilangan 22 palindrome namun dikarenakan dalam expresult menggunakan not palindrome sesuai dengan logika fungsi method Palindrome_1 dimana semua hasil test dalam palindrome akan eror jika tidak menggunakan NOT.

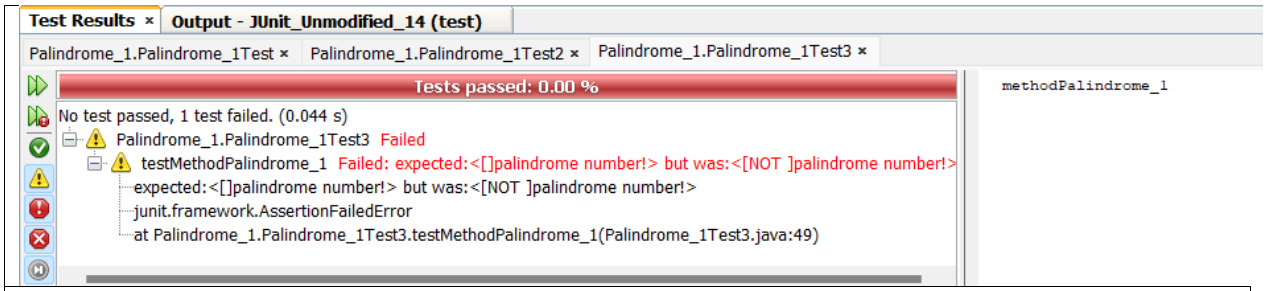
c. Input: 27

➢ **Snippet of test case**



```
16  *
17  * @author vanessa
18  */
19  public class Palindrome_1Test3 {
20
21      public Palindrome_1Test3() {...2 lines }
23
24      @BeforeClass
25      public static void setUpClass() {...2 lines }
27
28      @AfterClass
29      public static void tearDownClass() {...2 lines }
31
32      @Before
33      public void setUp() {...2 lines }
35
36      @After
37      public void tearDown() {...2 lines }
39      /**
40       * Test of methodPalindrome_1 method, of class Palindrome_1.
41       */
42      @Test
43      public void testMethodPalindrome_1() {
44          System.out.println("methodPalindrome_1");
45          int n1 = 27;
46          Palindrome_1 instance = new Palindrome_1();
47          String expResult = "palindrome number!";
48          String result = instance.methodPalindrome_1(n1);
49          assertEquals(expResult, result);
50          // TODO review the generated test code and remove the default call to fail.
51          //fail("The test case is a prototype.");
52      }
53
```

- **Snippet of results**

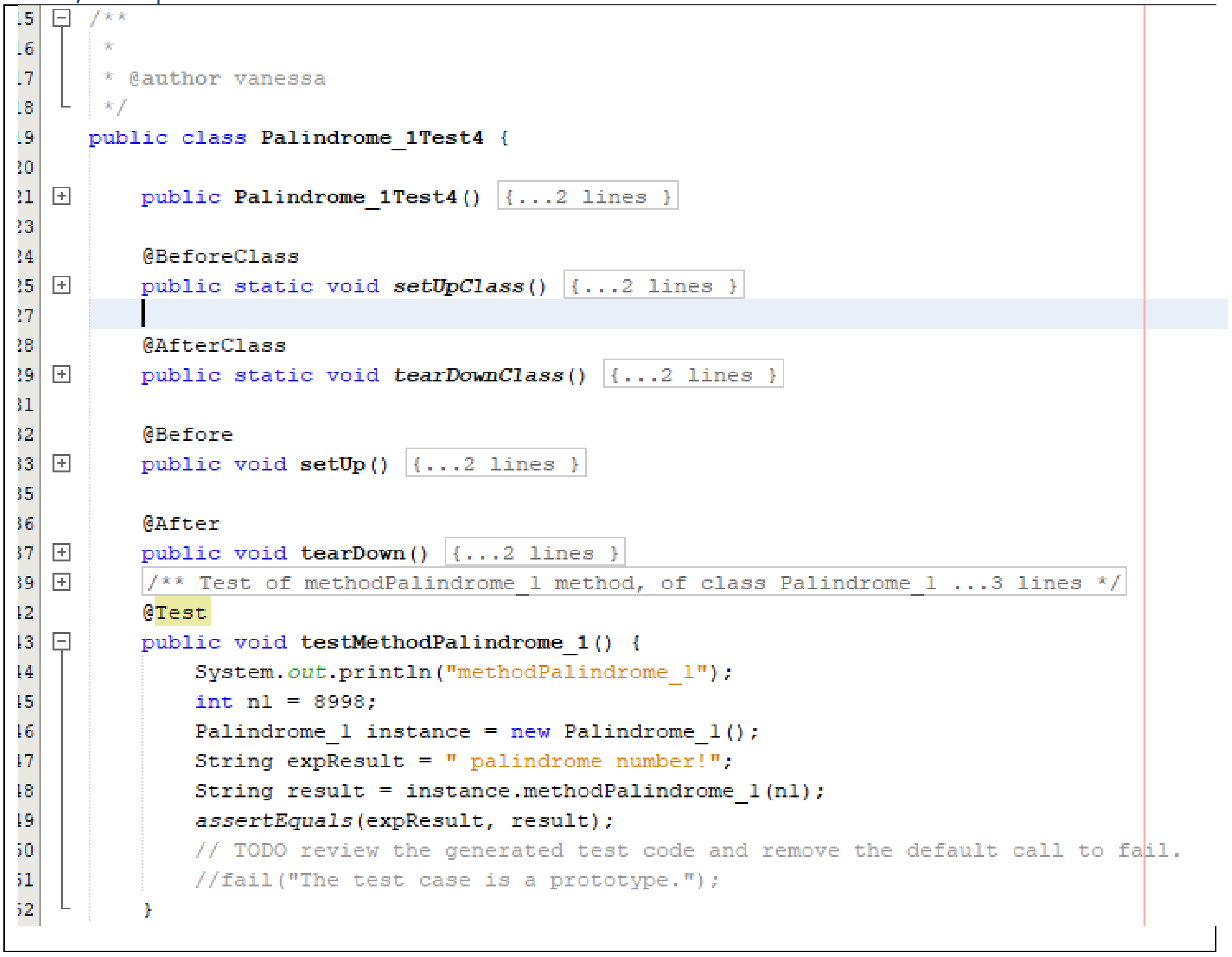


```
Test Results × | Output - JUnit_Unmodified_14 (test)
Palindrome_1.Palindrome_1Test × | Palindrome_1.Palindrome_1Test2 × | Palindrome_1.Palindrome_1Test3 ×
Tests passed: 0.00 %
No test passed, 1 test failed. (0.044 s)
  Palindrome_1.Palindrome_1Test3 Failed
    testMethodPalindrome_1 Failed: expected:<[]palindrome number!> but was:<[NOT ]palindrome number!>
      expected:<[]palindrome number!> but was:<[NOT ]palindrome number!>
      junit.framework.AssertionFailedError
      at Palindrome_1.Palindrome_1Test3.testMethodPalindrome_1(Palindrome_1Test3.java:49)

methodPalindrome_1
```

Penjelasan : Sebenarnya 27 merupakan not palindrome dikarenakan jika dibalikkan hasilnya 72 sangat beda, namun mengapa terjadi error dikarenakan memeodif maka pada line 44 agar test case testMethodPalindrome_1 berhasil, expResult harus diubah menjadi "NOT palindrome number!" sesuai dengan output yang diharapkan dari fungsi methodPalindrome_1 untuk bilangan.

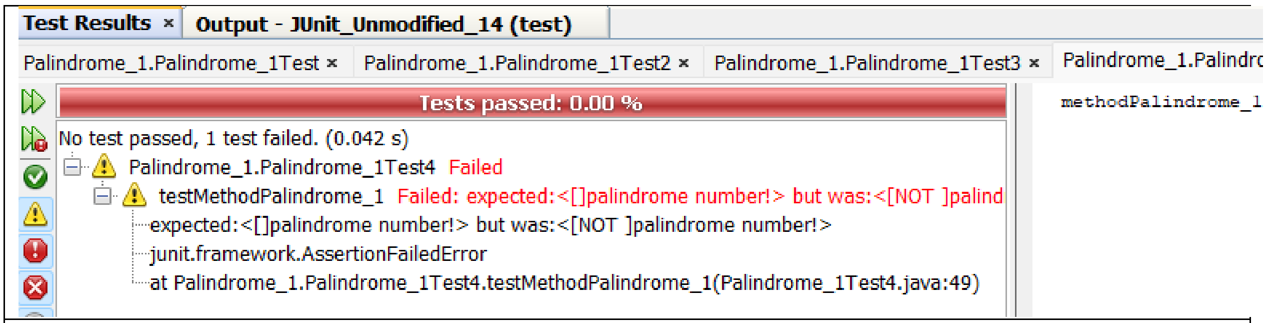
d. Input: 8998

- **Snippet of test case**



```
/**
 *
 * @author vanessa
 */
public class Palindrome_1Test4 {

    public Palindrome_1Test4() {...2 lines }

    @BeforeClass
    public static void setUpClass() {...2 lines }

    @AfterClass
    public static void tearDownClass() {...2 lines }

    @Before
    public void setUp() {...2 lines }

    @After
    public void tearDown() {...2 lines }
    /** Test of methodPalindrome_1 method, of class Palindrome_1 ...3 lines */
    @Test
    public void testMethodPalindrome_1() {
        System.out.println("methodPalindrome_1");
        int n1 = 8998;
        Palindrome_1 instance = new Palindrome_1();
        String expResult = " 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**

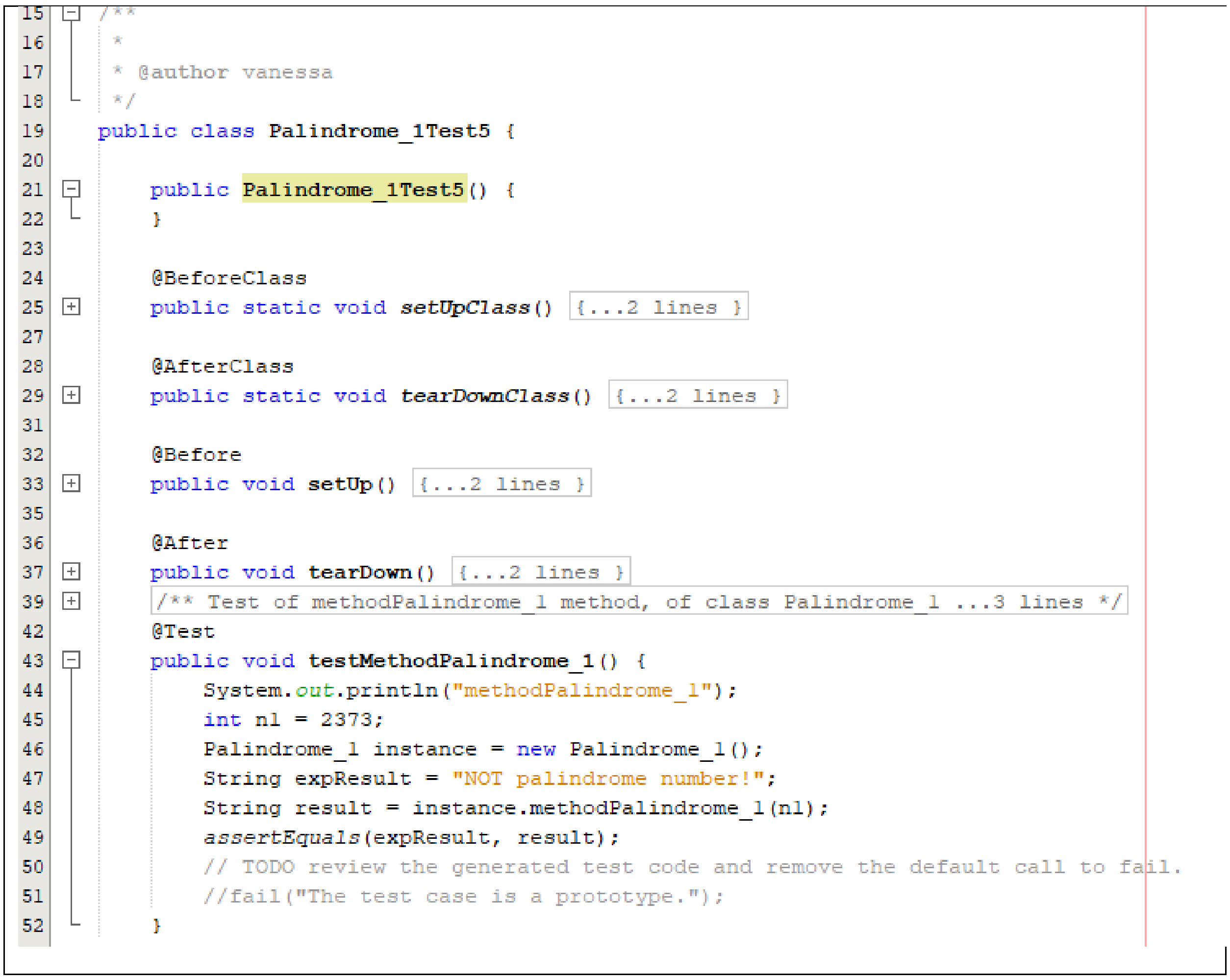


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
Palindrome_1.Palindrome_1Test ×   Palindrome_1.Palindrome_1Test2 ×   Palindrome_1.Palindrome_1Test3 ×   Palindrome_1.Palindrc

Tests passed: 0.00 %                                                              methodPalindrome_1
No test passed, 1 test failed. (0.042 s)
  Palindrome_1.Palindrome_1Test4  Failed
    testMethodPalindrome_1  Failed: expected:<[]palindrome number!> but was:<[NOT ]palind
      expected:<[]palindrome number!> but was:<[NOT ]palindrome number!>
      junit.framework.AssertionFailedError
      at Palindrome_1.Palindrome_1Test4.testMethodPalindrome_1(Palindrome_1Test4.java:49)
```

Penjelasan : Sebenarnya 8998 merupakan bilangan palindrome dikarenakan jika dibalikkan tetap hasilnya sama, namun mengapa terjadi error dikarenakan pada method menggunakan n1 = n1 * 10 seharusnya menggunakan n1 = n1 / 10 untuk menghindari perulangan tak terbatas dan memperbaiki perhitungan balik yang benar. Jika kita memeodif maka pada line 42 agar test case testMethodPalindrome_1 berhasil, expResult harus diubah menjadi "NOT palindrome number!" sesuai dengan output yang diharapkan dari fungsi methodPalindrome_1 untuk bilangan.

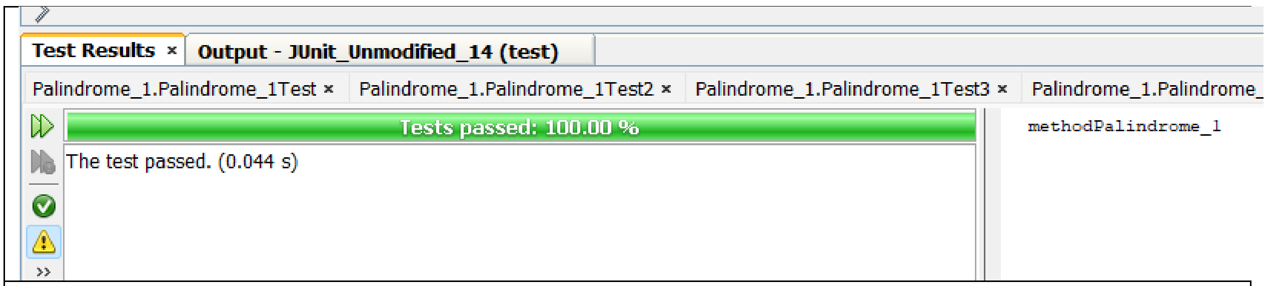
e. Input: 2373

➢ **Snippet of test case**



```
/**
 *
 * @author vanessa
 */
public class Palindrome_1Test5 {

    public Palindrome_1Test5() {
    }

    @BeforeClass
    public static void setUpClass() {...2 lines }

    @AfterClass
    public static void tearDownClass() {...2 lines }

    @Before
    public void setUp() {...2 lines }

    @After
    public void tearDown() {...2 lines }
    /** Test of methodPalindrome_1 method, of class Palindrome_1 ...3 lines */
    @Test
    public void testMethodPalindrome_1() {
        System.out.println("methodPalindrome_1");
        int n1 = 2373;
        Palindrome_1 instance = new Palindrome_1();
        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**

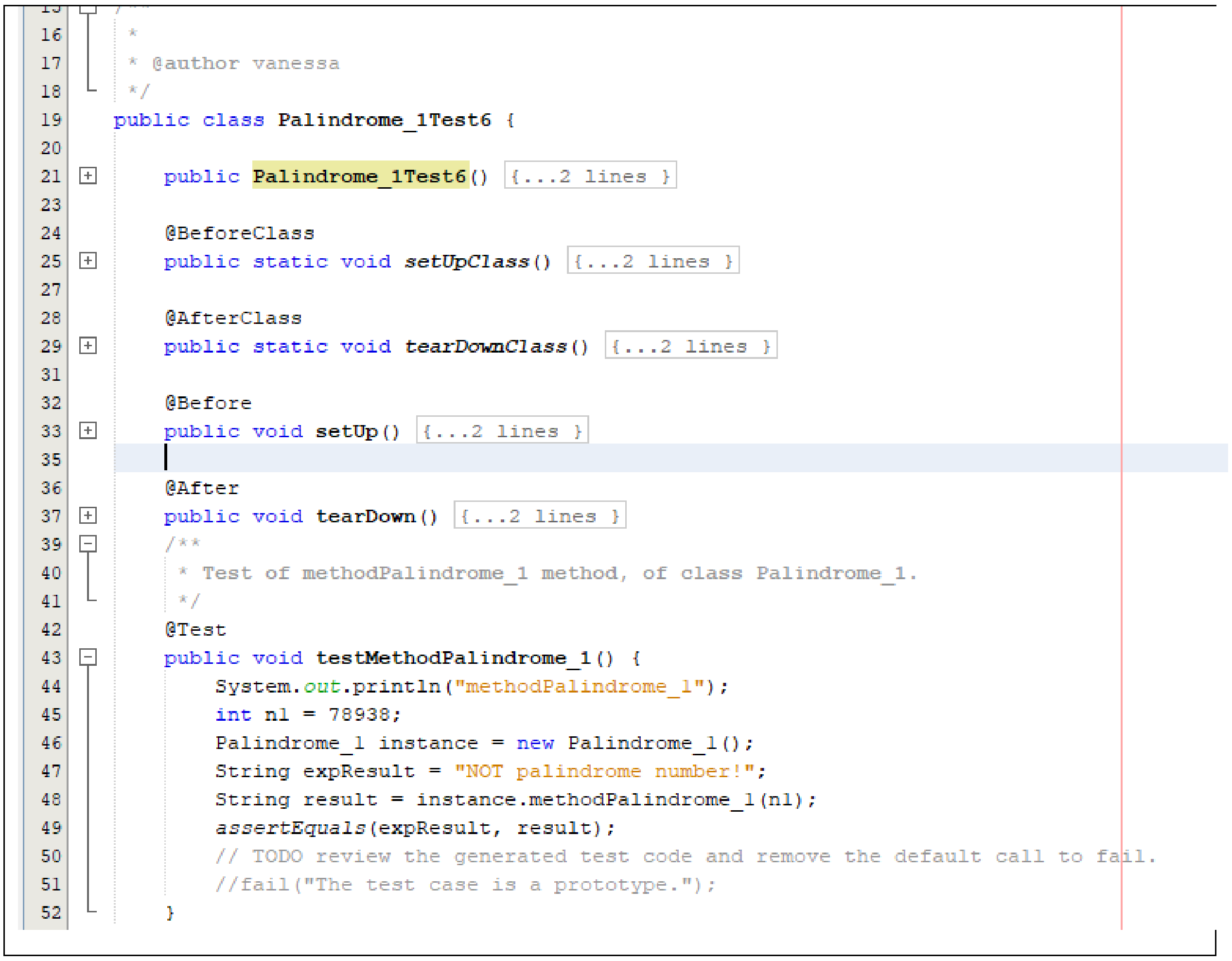


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
Palindrome_1.Palindrome_1Test ×   Palindrome_1.Palindrome_1Test2 ×   Palindrome_1.Palindrome_1Test3 ×   Palindrome_1.Palindrome_
Tests passed: 100.00 %
The test passed. (0.044 s)
methodPalindrome_1
```

Penjelasan : Sebenarnya 8998 merupakan bilangan palindrome dikarenakan jika dibalikkan tetap hasilnya sama, namun mengapa terjadi error dikarenakan pada method menggunakan n1 = n1 * 10 seharusnya menggunakan n1 = n1 / 10 untuk menghindari perulangan tak terbatas dan memperbaiki perhitungan balik yang benar. Jika kita memeodif maka pada line 42 agar test case testMethodPalindrome_1 berhasil, expResult harus diubah menjadi "NOT palindrome number!" sesuai dengan output yang diharapkan dari fungsi methodPalindrome_1 untuk bilangan .

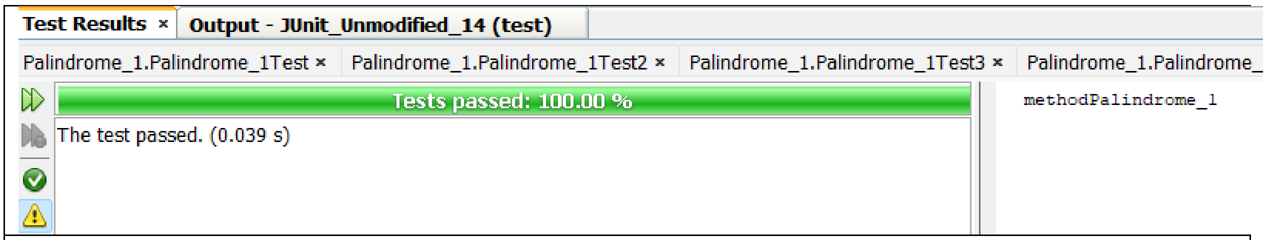
f. Input: 78938

➢ **Snippet of test case**



```
/**
 *
 * @author vanessa
 */
public class Palindrome_1Test6 {

    public Palindrome_1Test6() {...2 lines }

    @BeforeClass
    public static void setUpClass() {...2 lines }

    @AfterClass
    public static void tearDownClass() {...2 lines }

    @Before
    public void setUp() {...2 lines }

    @After
    public void tearDown() {...2 lines }
    /**
     * Test of methodPalindrome_1 method, of class Palindrome_1.
     */
    @Test
    public void testMethodPalindrome_1() {
        System.out.println("methodPalindrome_1");
        int n1 = 78938;
        Palindrome_1 instance = new Palindrome_1();
        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**



Test Results
Output - JUnit_Unmodified_14 (test)
Palindrome_1.Palindrome_1Test
Palindrome_1.Palindrome_1Test2
Palindrome_1.Palindrome_1Test3
Palindrome_1.Palindrome_

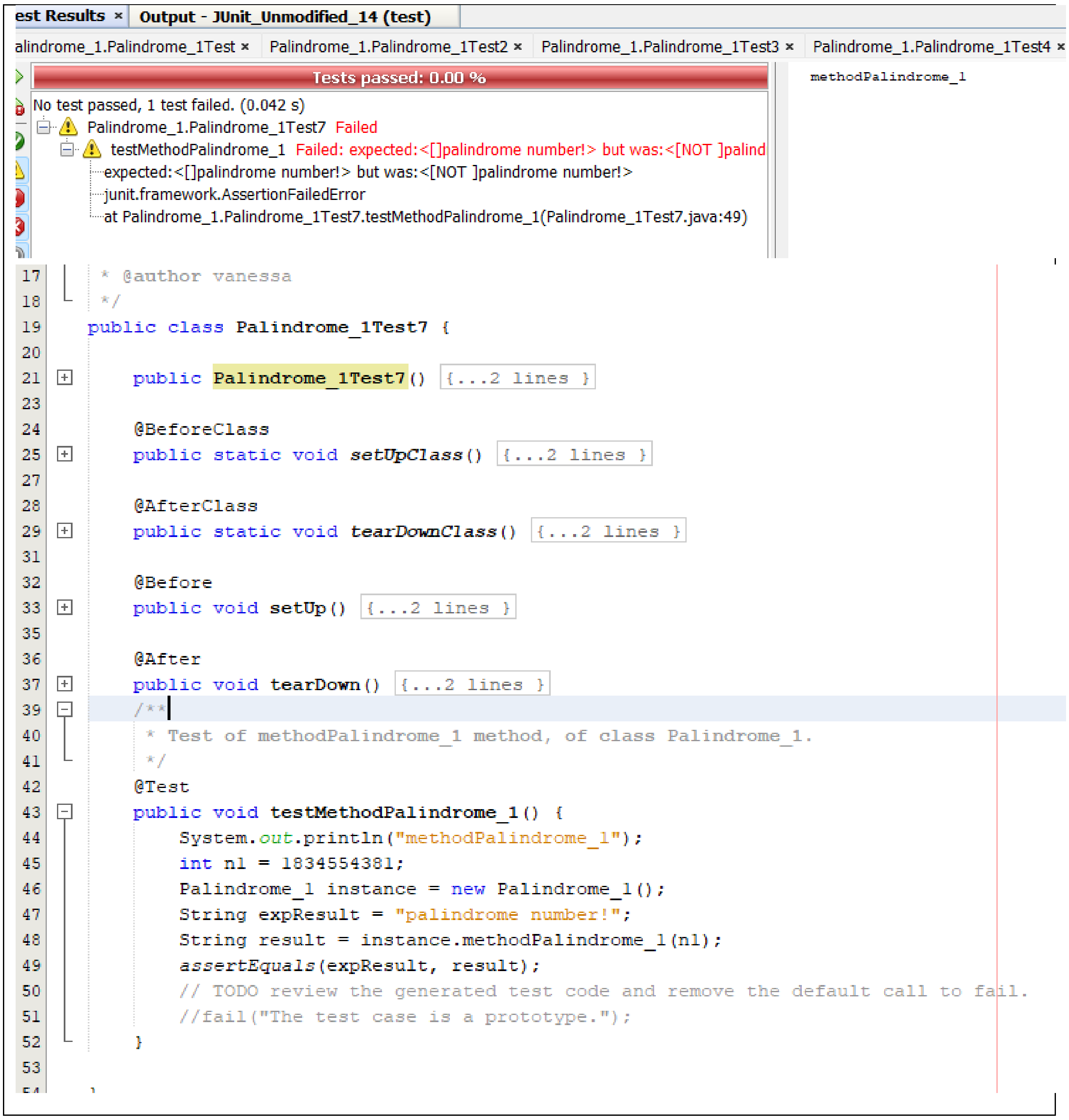
```
Tests passed: 100.00 %
The test passed. (0.039 s)
```

```
methodPalindrome_1
```

Penjelasan : Test berhasil bukan karena bilangan 78938 not palindrome namun dikarenakan dalam expresult menggunakan not palindrome sesuai dengan logika fungsi method Palindrome_1 dimana semua hasil test dalam palindrome akan eror jika tidak menggunakan NOT.

g. Input: 1834554381

➢ **Snippet of test case**



est Results × Output - JUnit_Unmodified_14 (test)

alindrome_1.Palindrome_1Test × Palindrome_1.Palindrome_1Test2 × Palindrome_1.Palindrome_1Test3 × Palindrome_1.Palindrome_1Test4 ×

Tests passed: 0.00 %

```
No test passed, 1 test failed. (0.042 s)
  Palindrome_1.Palindrome_1Test7  Failed
    testMethodPalindrome_1  Failed: expected:<[]palindrome number!> but was:<[NOT ]palind
      expected:<[]palindrome number!> but was:<[NOT ]palindrome number!>
      junit.framework.AssertionFailedError
      at Palindrome_1.Palindrome_1Test7.testMethodPalindrome_1(Palindrome_1Test7.java:49)
```

```
methodPalindrome_1
```

```
17   * @author vanessa
18   */
19  public class Palindrome_1Test7 {
20  
21      public Palindrome_1Test7() {...2 lines }
23  
24      @BeforeClass
25      public static void setUpClass() {...2 lines }
27  
28      @AfterClass
29      public static void tearDownClass() {...2 lines }
31  
32      @Before
33      public void setUp() {...2 lines }
35  
36      @After
37      public void tearDown() {...2 lines }
39      /**
40       * Test of methodPalindrome_1 method, of class Palindrome_1.
41       */
42      @Test
43      public void testMethodPalindrome_1() {
44          System.out.println("methodPalindrome_1");
45          int n1 = 1834554381;
46          Palindrome_1 instance = new Palindrome_1();
47          String expResult = "palindrome number!";
48          String result = instance.methodPalindrome_1(n1);
49          assertEquals(expResult, result);
50          // TODO review the generated test code and remove the default call to fail.
51          //fail("The test case is a prototype.");
52      }
53  
54  }
```

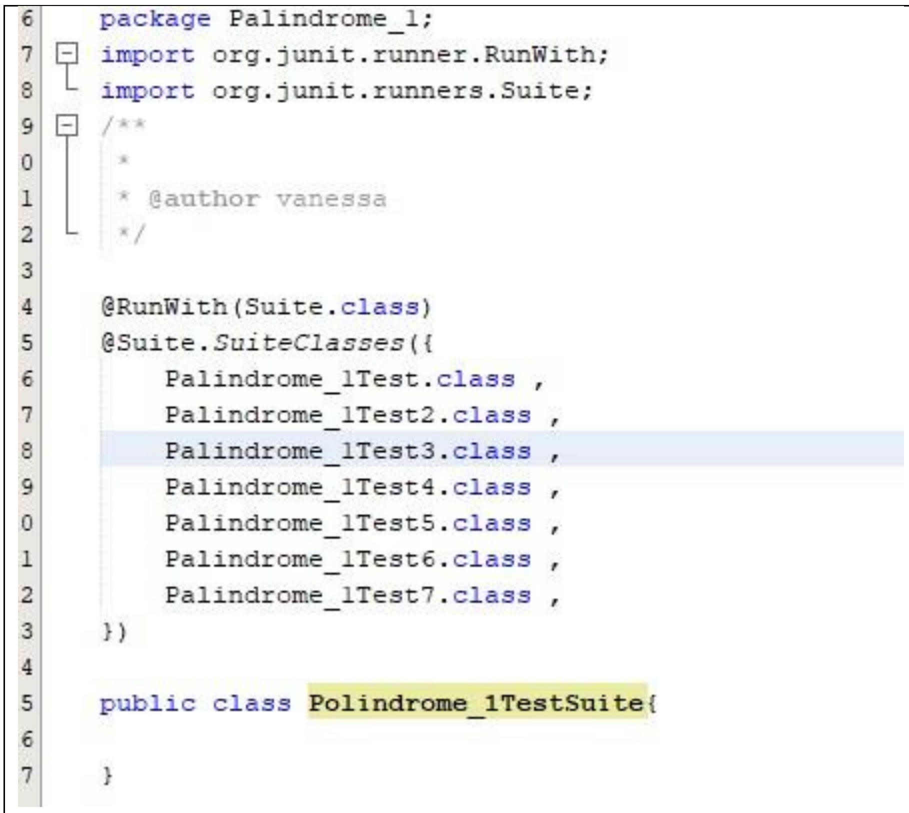
➢ **Snippet of results**

Penjelasan : Sebenarnya 1834554381 merupakan bilangan palindrome dikarenakan jika dibalikkan tetap hasilnya sama, namun mengapa terjadi error dikarenakan pada method menggunakan n1 = n1 * 10 seharusnya menggunakan n1 = n1 / 10 untuk menghindari perulangan tak terbatas dan memperbaiki perhitungan balik yang benar. Jika kita memeodif maka pada line 42 agar test case testMethodPalindrome_1 berhasil, expResult harus diubah

menjadi "NOT palindrome number!" sesuai dengan output yang diharapkan dari fungsi methodPalindrome_1 untuk bilangan..

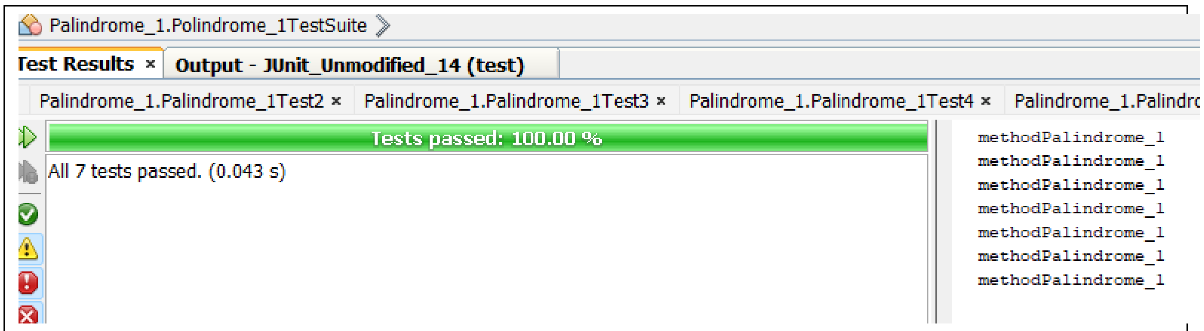
h. TestSuite

- **Snippet of test case**



```
package Palindrome_1;
import org.junit.runner.RunWith;
import org.junit.runners.Suite;
/**
 *
 * @author vanessa
 */

@RunWith(Suite.class)
@Suite.SuiteClasses({
    Palindrome_1Test.class ,
    Palindrome_1Test2.class ,
    Palindrome_1Test3.class ,
    Palindrome_1Test4.class ,
    Palindrome_1Test5.class ,
    Palindrome_1Test6.class ,
    Palindrome_1Test7.class ,
})

public class Polindrome_1TestSuite{

}
```

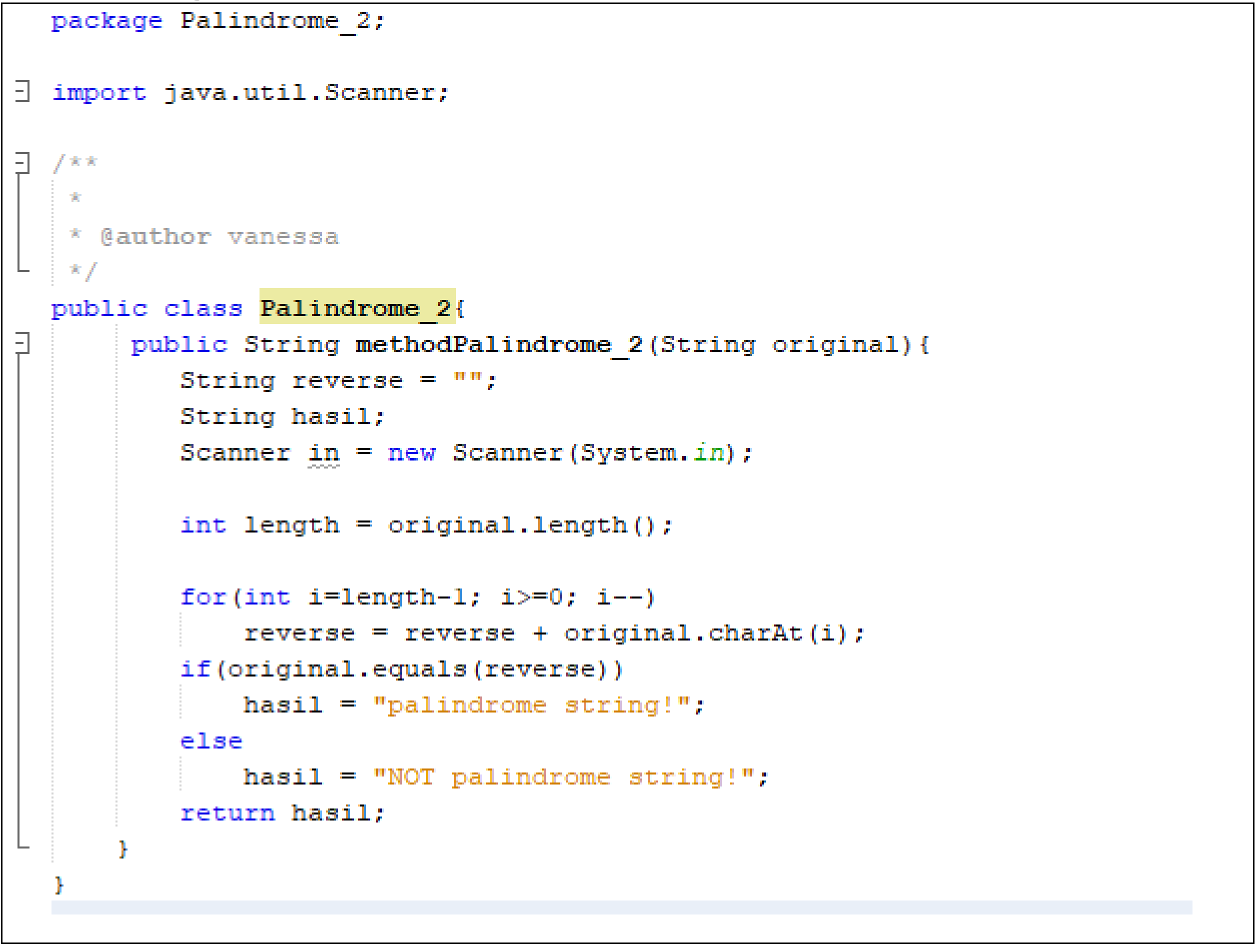
- **Snippet of results**



```
Palindrome_1.Polindrome_1TestSuite
Test Results × | Output - JUnit_Unmodified_14 (test)
Palindrome_1.Palindrome_1Test2 × | Palindrome_1.Palindrome_1Test3 × | Palindrome_1.Palindrome_1Test4 × | Palindrome_1.Palindrc
Tests passed: 100.00 %
All 7 tests passed. (0.043 s)

methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
```

Penjelasan : Sebelumnya semua test eror, namun kami menjalankan sesuai dengan arahan dalam konteks makanya tidak ada test yang error dikarena semua berhasil dengan memenuhi yang ada di pesan message di test test sebelumnya.

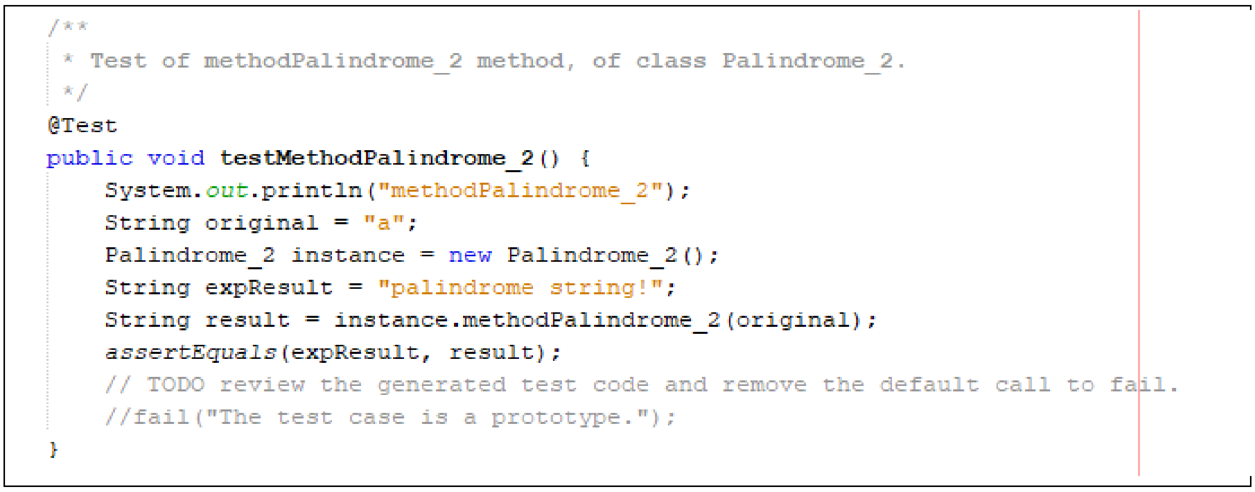
2) Testing for Palindrome_2.java



```java
package Palindrome_2;

import java.util.Scanner;

/**
 *
 * @author vanessa
 */
public class Palindrome_2{
        public String methodPalindrome_2(String original){
            String reverse = "";
            String hasil;
            Scanner in = new Scanner(System.in);

            int length = original.length();

            for(int i=length-1; i>=0; i--)
                reverse = reverse + original.charAt(i);
            if(original.equals(reverse))
                hasil = "palindrome string!";
            else
                hasil = "NOT palindrome string!";
            return hasil;
        }
}
```

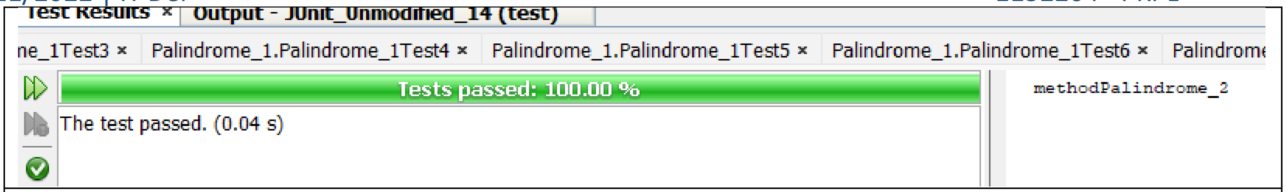
a. Input: a

- ➢ **Snippet of test case**



```java
/**
 * Test of methodPalindrome_2 method, of class Palindrome_2.
 */
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "a";
    Palindrome_2 instance = new Palindrome_2();
    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**

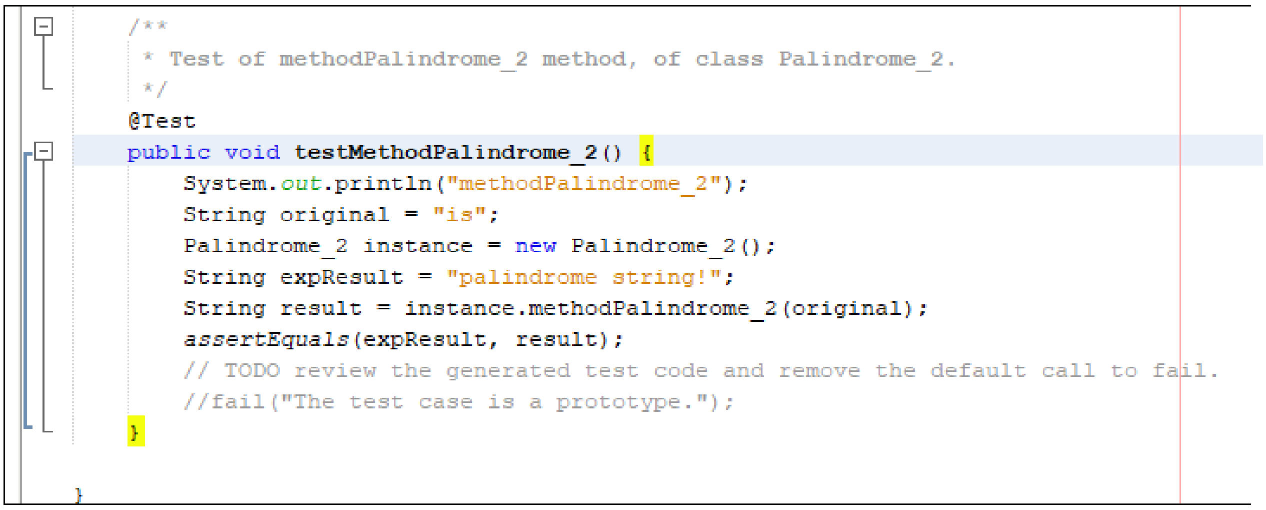


```
Test Results × | Output - JUnit_Unmodified_14 (test)
ne_1Test3 × | Palindrome_1.Palindrome_1Test4 × | Palindrome_1.Palindrome_1Test5 × | Palindrome_1.Palindrome_1Test6 × | Palindrome
Tests passed: 100.00 %                                   methodPalindrome_2
The test passed. (0.04 s)
```

Penjelasan : string "a"merupakan palindrome karena string ini hanya terdiri dari satu karakter, maka secara otomatis string ini adalah palindrome. Ketika membalikkan string "a", hasilnya tetap "a", sehingga string asli dan string yang sudah dibalikkan sama, dan metode methodPalindrome_2 akan mengembalikan hasil "palindrome string.

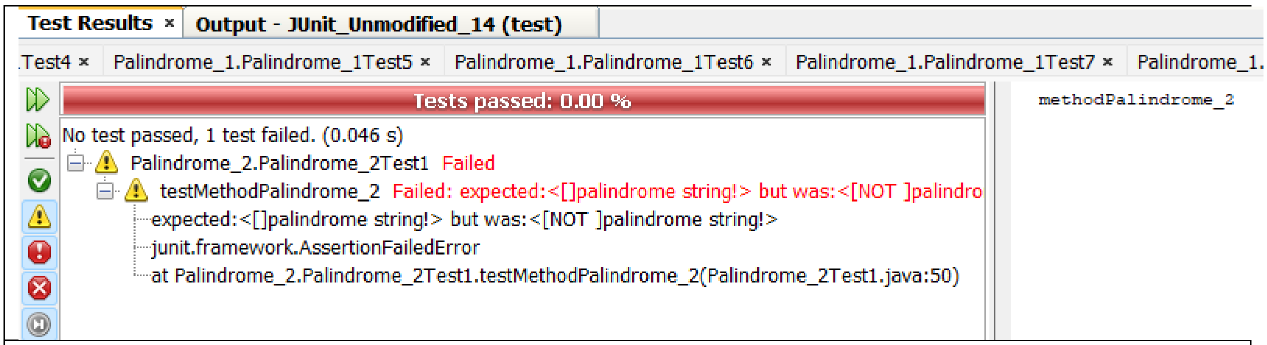
b. Input: is

➢ **Snippet of test case**



```java
    /**
     * Test of methodPalindrome_2 method, of class Palindrome_2.
     */
    @Test
    public void testMethodPalindrome_2() {
        System.out.println("methodPalindrome_2");
        String original = "is";
        Palindrome_2 instance = new Palindrome_2();
        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**

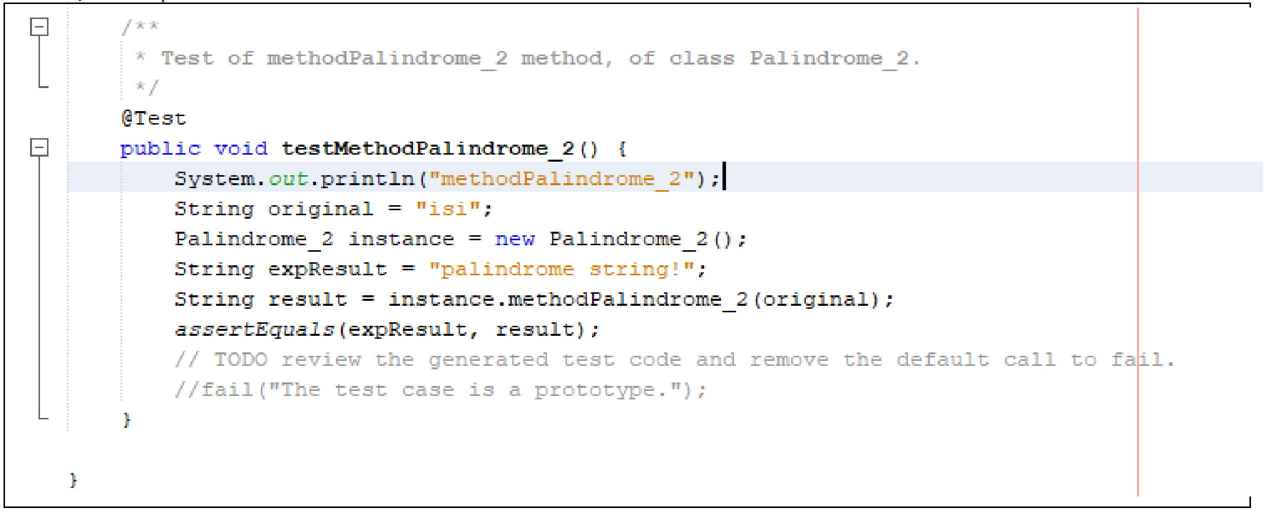


```
Test Results × | Output - JUnit_Unmodified_14 (test)
Test4 × | Palindrome_1.Palindrome_1Test5 × | Palindrome_1.Palindrome_1Test6 × | Palindrome_1.Palindrome_1Test7 × | Palindrome_1.
Tests passed: 0.00 %                                     methodPalindrome_2
No test passed, 1 test failed. (0.046 s)
  Palindrome_2.Palindrome_2Test1  Failed
    testMethodPalindrome_2  Failed: expected:<[]palindrome string!> but was:<[NOT ]palindro
      expected:<[]palindrome string!> but was:<[NOT ]palindrome string!>
      junit.framework.AssertionFailedError
      at Palindrome_2.Palindrome_2Test1.testMethodPalindrome_2(Palindrome_2Test1.java:50)
```

Penjelasan : Test tidak berhasil dikarenakan string”is”jika dibalikan menjadi “si” yang diminta diatas adalah NOT palindrome.

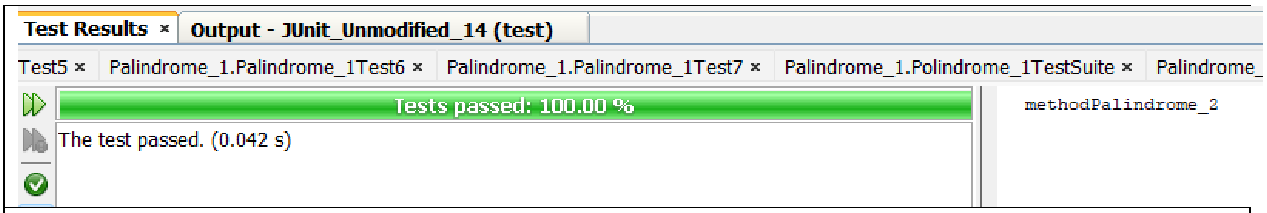
c. Input: isi

➢ **Snippet of test case**



```
    /**
     * Test of methodPalindrome_2 method, of class Palindrome_2.
     */
    @Test
    public void testMethodPalindrome_2() {
        System.out.println("methodPalindrome_2");
        String original = "isi";
        Palindrome_2 instance = new Palindrome_2();
        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**

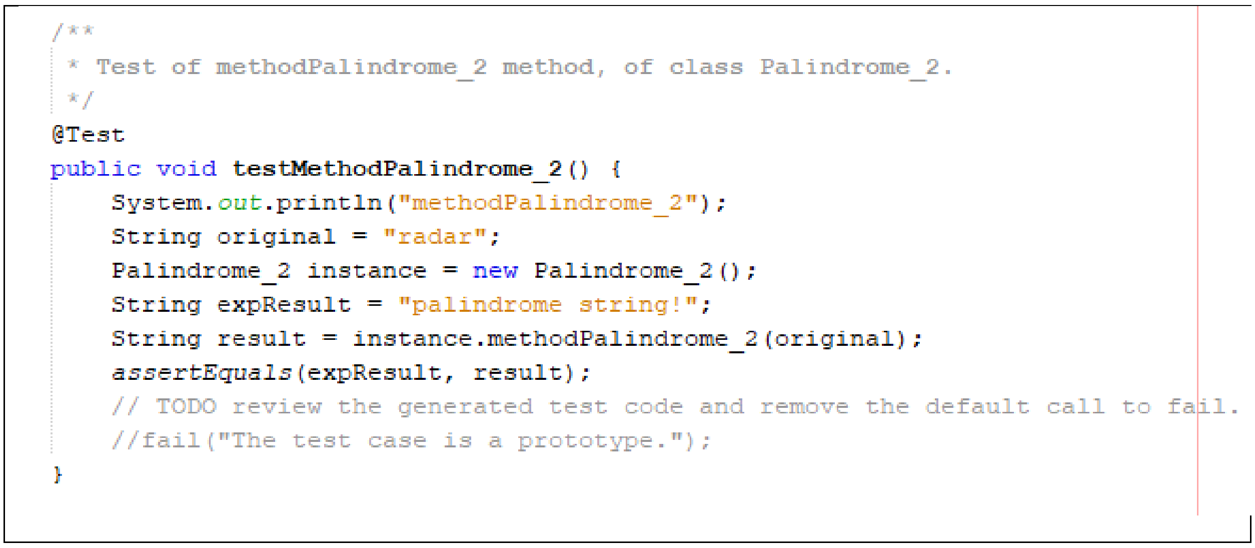


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
Test5 ×   Palindrome_1.Palindrome_1Test6 ×   Palindrome_1.Palindrome_1Test7 ×   Palindrome_1.Polindrome_1TestSuite ×   Palindrome_
Tests passed: 100.00 %
The test passed. (0.042 s)
methodPalindrome_2
```

Penjelasan : string "isi"merupakan palindrome karena string ini hanya terdiri dari satu karakter, maka secara otomatis string ini adalah palindrome. Ketika membalikkan string "isi", hasilnya tetap "isi", sehingga string asli dan string yang sudah dibalikkan sama, dan metode methodPalindrome_2 akan mengembalikan hasil "palindrome string.

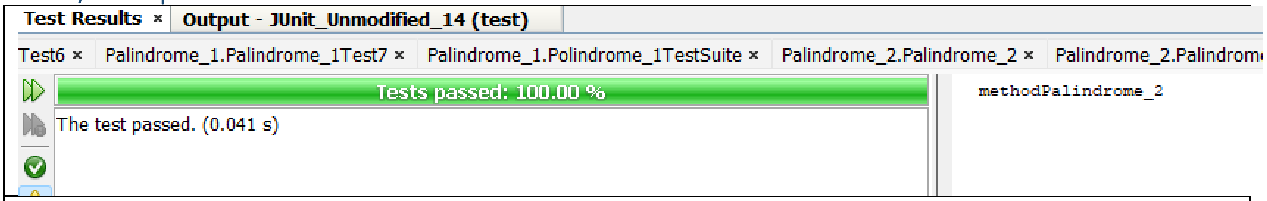
d. Input: radar

➢ **Snippet of test case**



```
/**
 * Test of methodPalindrome_2 method, of class Palindrome_2.
 */
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "radar";
    Palindrome_2 instance = new Palindrome_2();
    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**

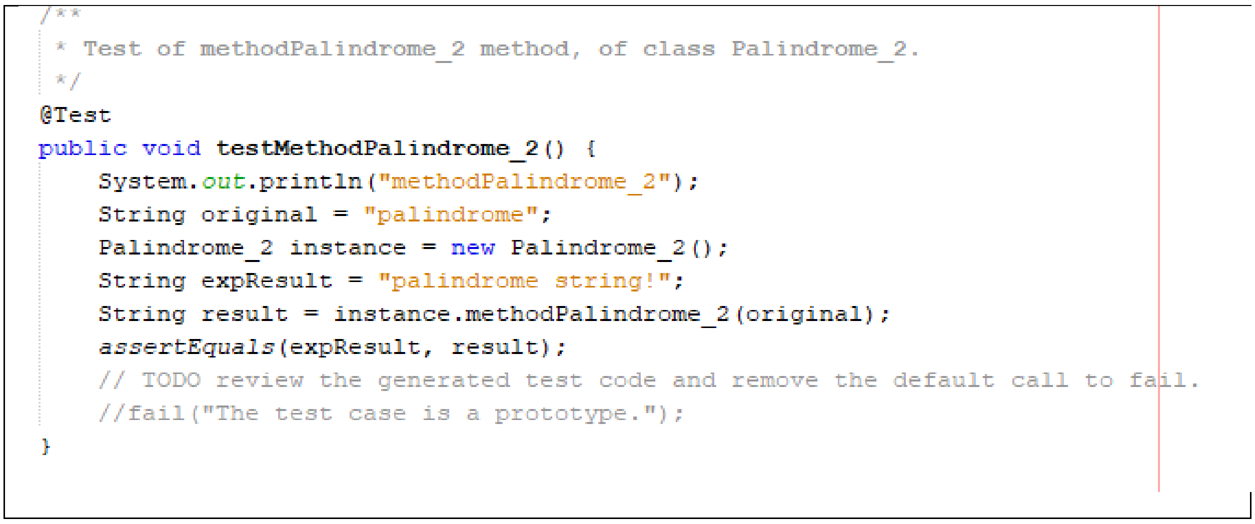


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
Test6 ×   Palindrome_1.Palindrome_1Test7 ×   Palindrome_1.Polindrome_1TestSuite ×   Palindrome_2.Palindrome_2 ×   Palindrome_2.Palindrom
Tests passed: 100.00 %                                   methodPalindrome_2
The test passed. (0.041 s)
```

Penjelasan : string "radar"merupakan palindrome karena string ini hanya terdiri dari satu karakter, maka secara otomatis string ini adalah palindrome. Ketika membalikkan string "radar", hasilnya tetap "radar", sehingga string asli dan string yang sudah dibalikkan sama, dan metode methodPalindrome_2 akan mengembalikan hasil "palindrome string.

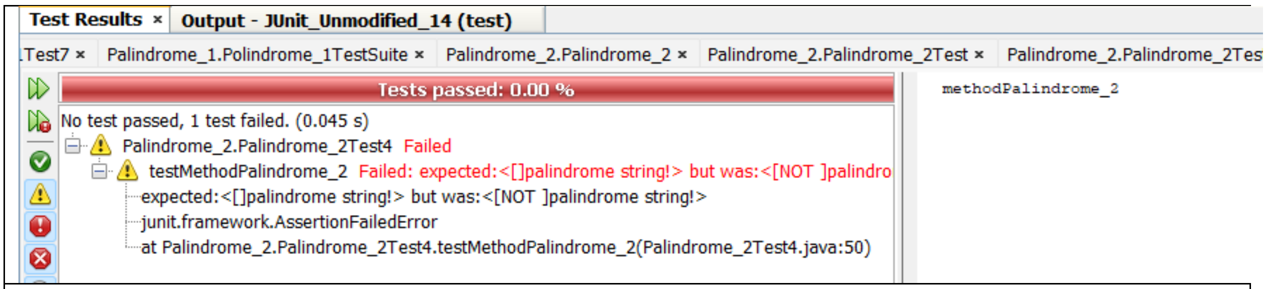
e. Input: palindrome

➢ **Snippet of test case**



```java
/**
 * Test of methodPalindrome_2 method, of class Palindrome_2.
 */
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "palindrome";
    Palindrome_2 instance = new Palindrome_2();
    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**

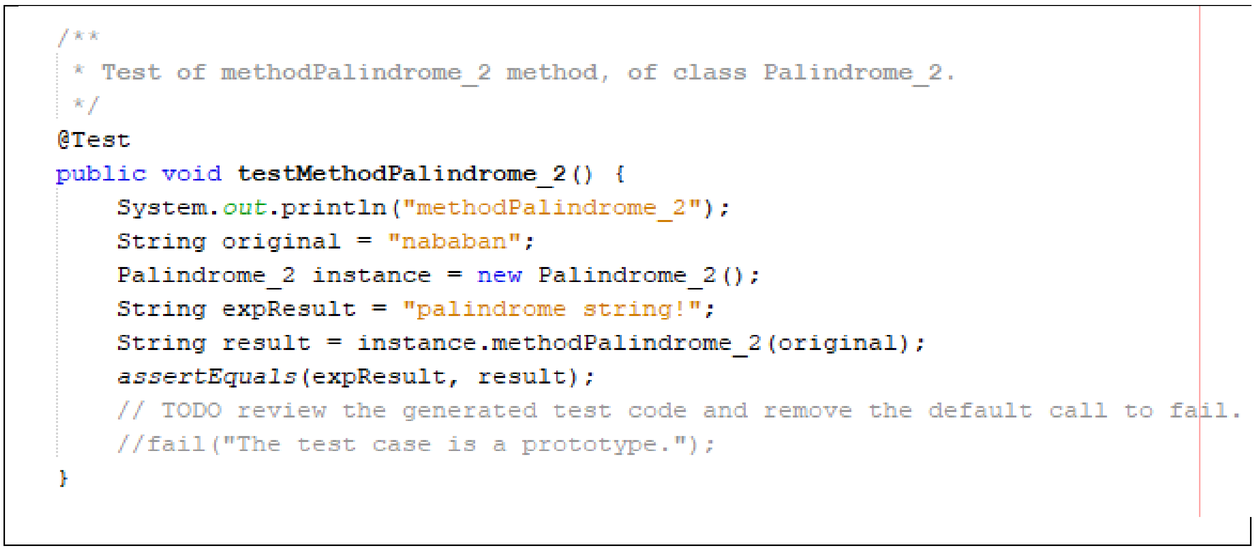


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
Test7 ×   Palindrome_1.Polindrome_1TestSuite ×   Palindrome_2.Palindrome_2 ×   Palindrome_2.Palindrome_2Test ×   Palindrome_2.Palindrome_2Tes
Tests passed: 0.00 %                                   methodPalindrome_2
No test passed, 1 test failed. (0.045 s)
  Palindrome_2.Palindrome_2Test4  Failed
    testMethodPalindrome_2  Failed: expected:<[]palindrome string!> but was:<[NOT ]palindro
      expected:<[]palindrome string!> but was:<[NOT ]palindrome string!>
      junit.framework.AssertionFailedError
      at Palindrome_2.Palindrome_2Test4.testMethodPalindrome_2(Palindrome_2Test4.java:50)
```

Penjelasan : Test tidak berhasil dikarenakan string"palindrome"jika dibalikan menjadi "emordnilap" yang diminta diatas adalah NOT palindrome.

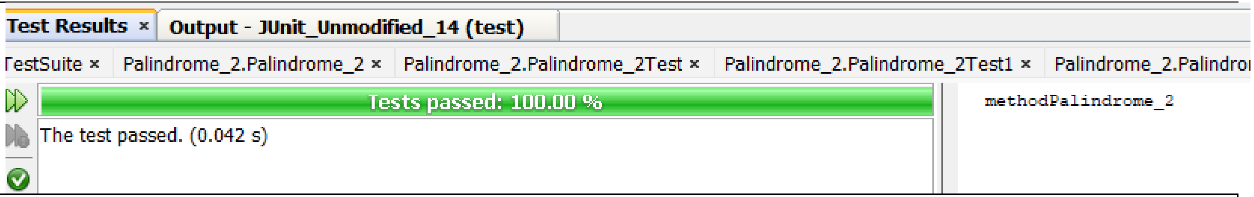
f. Input: nababan

➢ **Snippet of test case**



```
/**
 * Test of methodPalindrome_2 method, of class Palindrome_2.
 */
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "nababan";
    Palindrome_2 instance = new Palindrome_2();
    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**



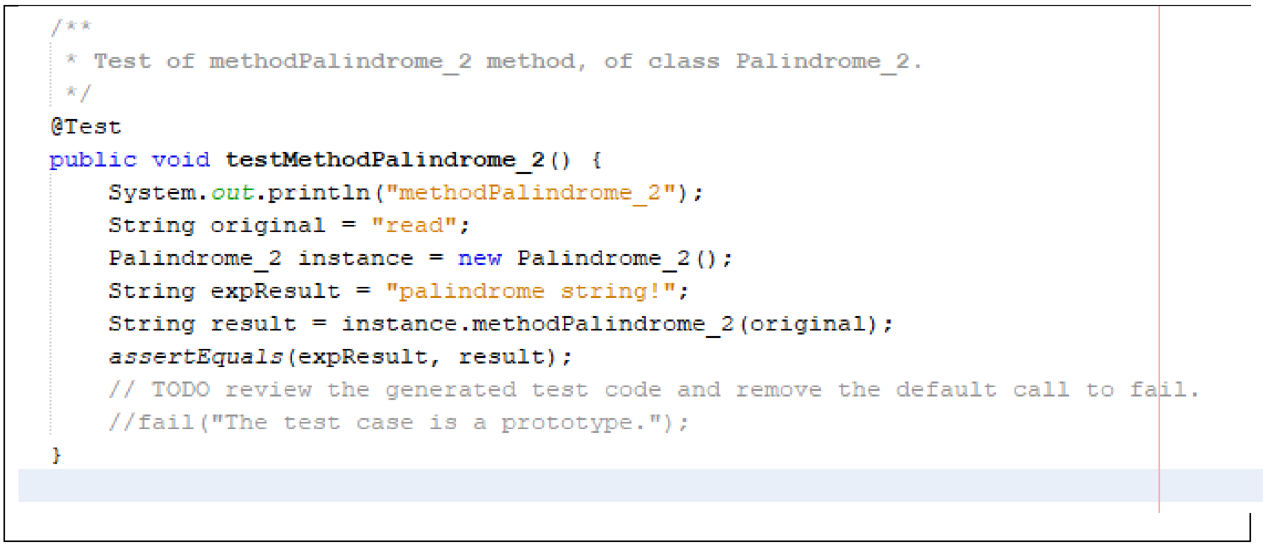
Test Results ×
Output - JUnit_Unmodified_14 (test)
TestSuite ×
Palindrome_2.Palindrome_2 ×
Palindrome_2.Palindrome_2Test ×
Palindrome_2.Palindrome_2Test1 ×
Palindrome_2.Palindro

```
Tests passed: 100.00 %
The test passed. (0.042 s)
methodPalindrome_2
```

Penjelasan : string "nababan"merupakan palindrome karena string ini hanya terdiri dari satu karakter, maka secara otomatis string ini adalah nababan. Ketika membalikkan string "nababan", hasilnya tetap "nababan", sehingga string asli dan string yang sudah dibalikkan sama, dan metode methodPalindrome_2 akan mengembalikan hasil "palindrome string.

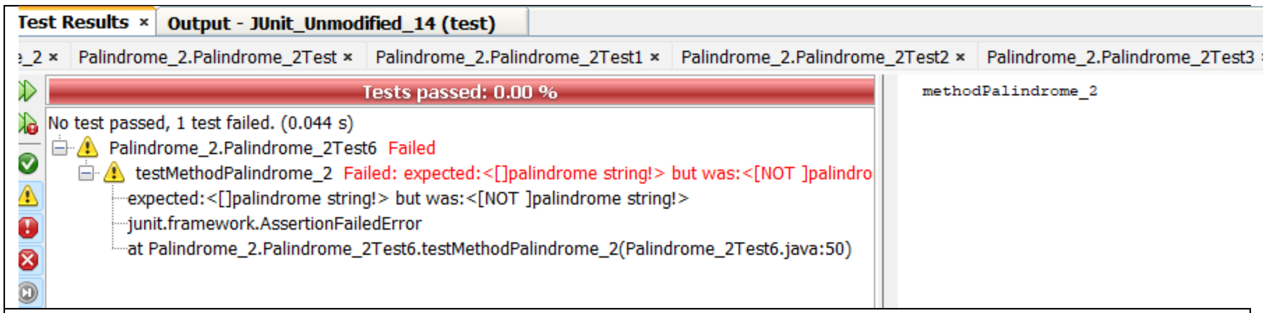
g. Input: read

- **Snippet of test case**



```
/**
 * Test of methodPalindrome_2 method, of class Palindrome_2.
 */
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "read";
    Palindrome_2 instance = new Palindrome_2();
    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**

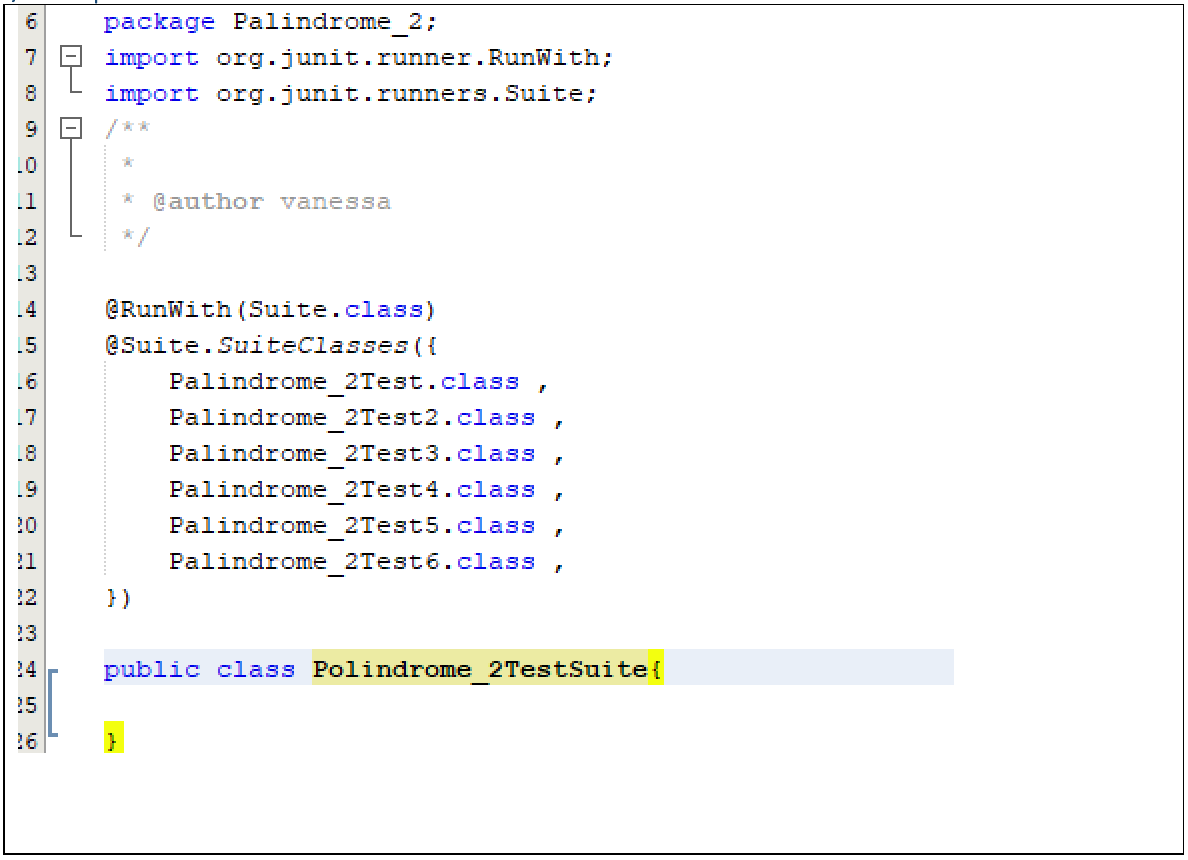


```
Test Results ×  Output - JUnit_Unmodified_14 (test)
_2 ×  Palindrome_2.Palindrome_2Test ×  Palindrome_2.Palindrome_2Test1 ×  Palindrome_2.Palindrome_2Test2 ×  Palindrome_2.Palindrome_2Test3
Tests passed: 0.00 %                                              methodPalindrome_2
No test passed, 1 test failed. (0.044 s)
  Palindrome_2.Palindrome_2Test6  Failed
    testMethodPalindrome_2  Failed: expected:<[]palindrome string!> but was:<[NOT ]palindro
      expected:<[]palindrome string!> but was:<[NOT ]palindrome string!>
      junit.framework.AssertionFailedError
      at Palindrome_2.Palindrome_2Test6.testMethodPalindrome_2(Palindrome_2Test6.java:50)
```

Penjelasan : Test tidak berhasil dikarenakan string"read"jika dibalikan menjadi "daer" yang diminta diatas adalah NOT palindrome.

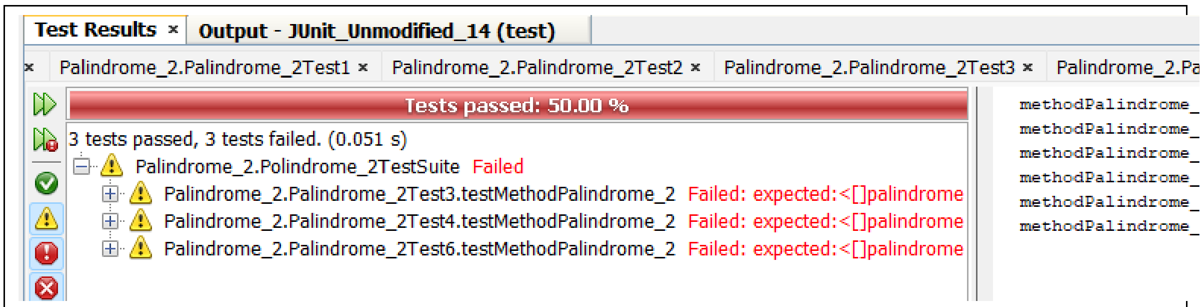
h. TestSuite

- **Snippet of test case**



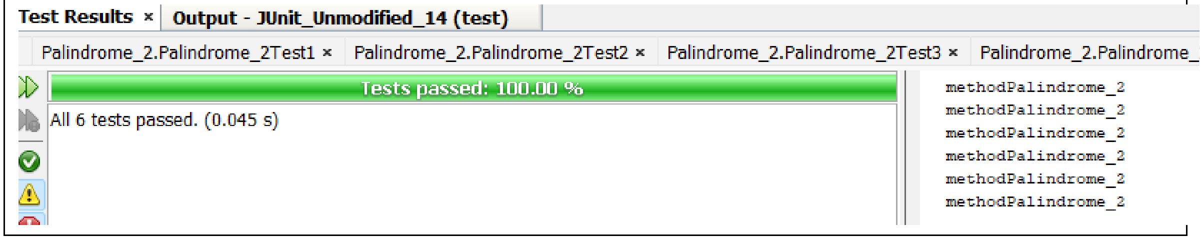
```
package Palindrome_2;
import org.junit.runner.RunWith;
import org.junit.runners.Suite;
/**
 *
 * @author vanessa
 */

@RunWith(Suite.class)
@Suite.SuiteClasses({
    Palindrome_2Test.class ,
    Palindrome_2Test2.class ,
    Palindrome_2Test3.class ,
    Palindrome_2Test4.class ,
    Palindrome_2Test5.class ,
    Palindrome_2Test6.class ,
})

public class Polindrome_2TestSuite{

}
```

➢ **Snippet of results**



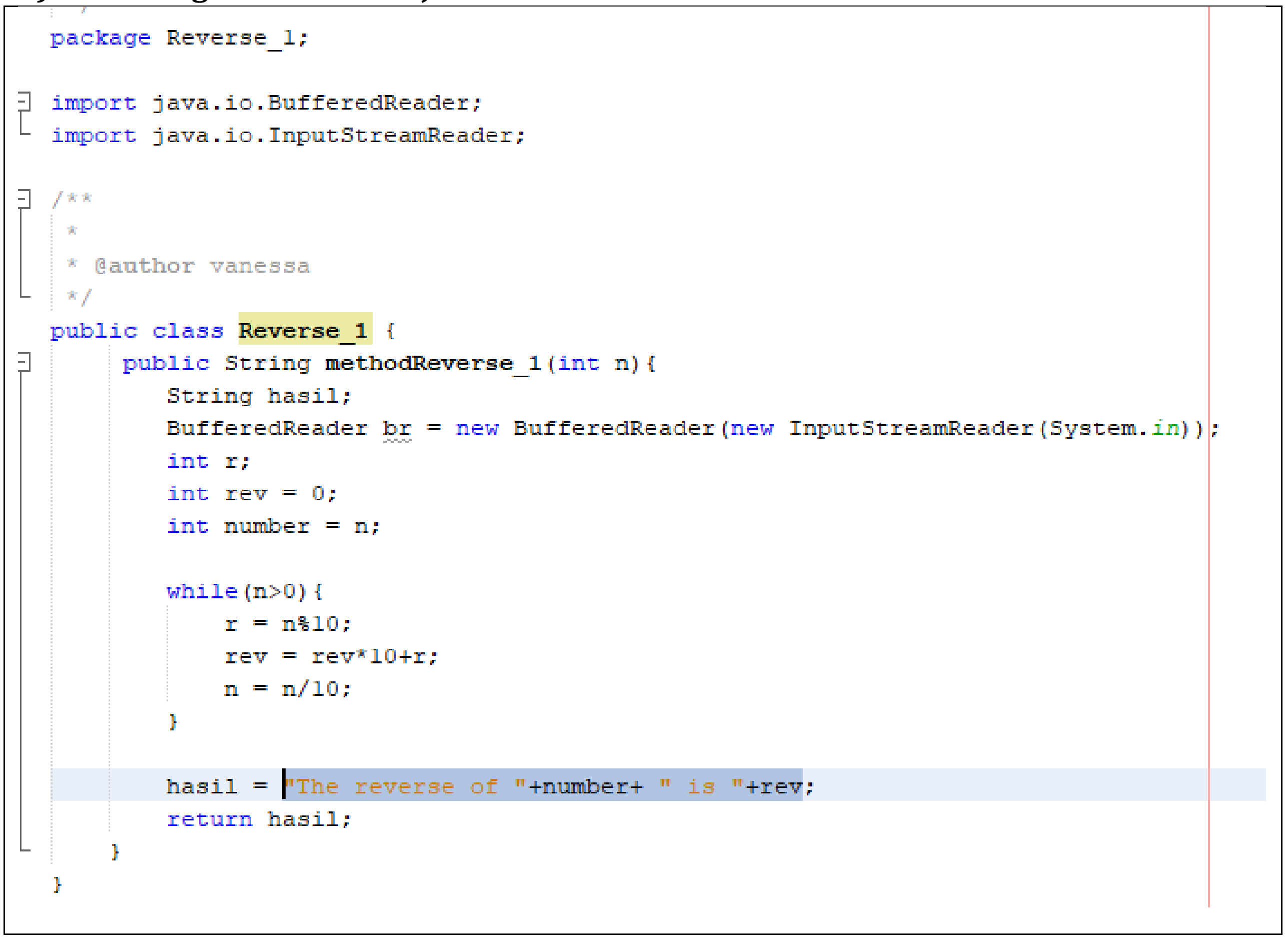
```
Test Results × | Output - JUnit_Unmodified_14 (test)
× Palindrome_2.Palindrome_2Test1 × Palindrome_2.Palindrome_2Test2 × Palindrome_2.Palindrome_2Test3 × Palindrome_2.Pa
Tests passed: 50.00 %
3 tests passed, 3 tests failed. (0.051 s)
  Palindrome_2.Polindrome_2TestSuite Failed
    Palindrome_2.Palindrome_2Test3.testMethodPalindrome_2 Failed: expected:<[]palindrome
    Palindrome_2.Palindrome_2Test4.testMethodPalindrome_2 Failed: expected:<[]palindrome
    Palindrome_2.Palindrome_2Test6.testMethodPalindrome_2 Failed: expected:<[]palindrome
methodPalindrome_
methodPalindrome_
methodPalindrome_
methodPalindrome_
methodPalindrome_
methodPalindrome_
```

Penjelasan : Terdapat beberapa eror dikarena ada yang tidak sesuai, namun dalam eror sudah dairahakan untuk kesesuain maka hasilnya akan menjadi seperti tapi hasilnya akan seperti dibawa ini



```
Test Results × | Output - JUnit_Unmodified_14 (test)
Palindrome_2.Palindrome_2Test1 × Palindrome_2.Palindrome_2Test2 × Palindrome_2.Palindrome_2Test3 × Palindrome_2.Palindrome_
Tests passed: 100.00 %
All 6 tests passed. (0.045 s)
methodPalindrome_2
methodPalindrome_2
methodPalindrome_2
methodPalindrome_2
methodPalindrome_2
methodPalindrome_2
```

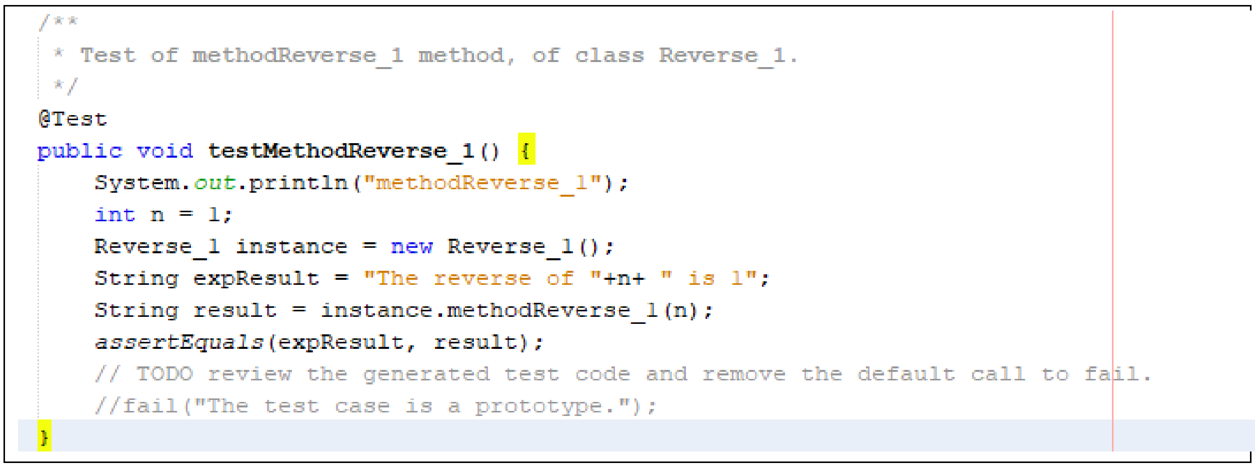
3) Testing for Reverse_1.java



```java
package Reverse_1;

import java.io.BufferedReader;
import java.io.InputStreamReader;

/**
 *
 * @author vanessa
 */
public class Reverse_1 {
    public String methodReverse_1(int n){
        String hasil;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int r;
        int rev = 0;
        int number = n;

        while(n>0){
            r = n%10;
            rev = rev*10+r;
            n = n/10;
        }

        hasil = "The reverse of "+number+ " is "+rev;
        return hasil;
    }
}
```

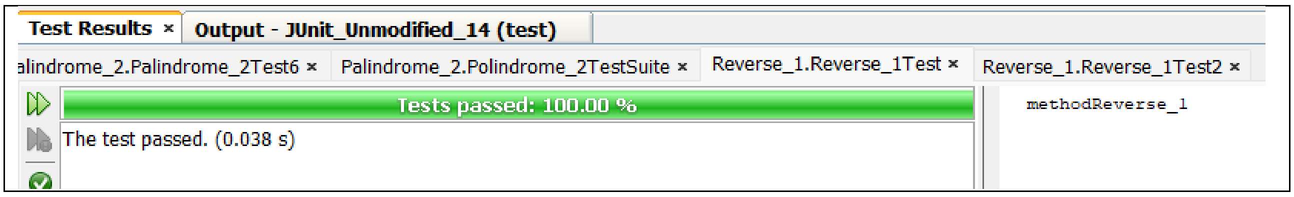
i. Input: 1

➢ **Snippet of test case**



```java
/**
 * Test of methodReverse_1 method, of class Reverse_1.
 */
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 1;
    Reverse_1 instance = new Reverse_1();
    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**



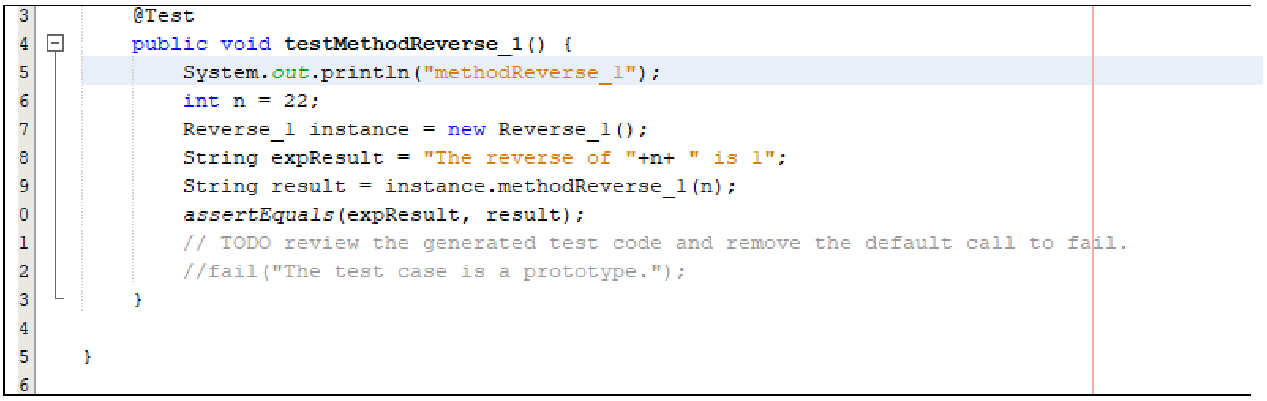
Test Results
Output - JUnit_Unmodified_14 (test)
alindrome_2.Palindrome_2Test6
Palindrome_2.Polindrome_2TestSuite
Reverse_1.Reverse_1Test
Reverse_1.Reverse_1Test2

```
Tests passed: 100.00 %
The test passed. (0.038 s)
methodReverse_1
```

Penjelasan : Test ini berhasil karena string "The reverse of 1 is 1" sesuai dengan hasil yang diharapkan yang Anda tentukan dalam variabel expResult. Saat Anda memanggil methodReverse_1(1), metode tersebut menghitung nilai balik dari bilangan 1, yang tetap sama yaitu 1. Sehingga, hasil yang dikembalikan oleh metode adalah "The reverse of 1 is 1.

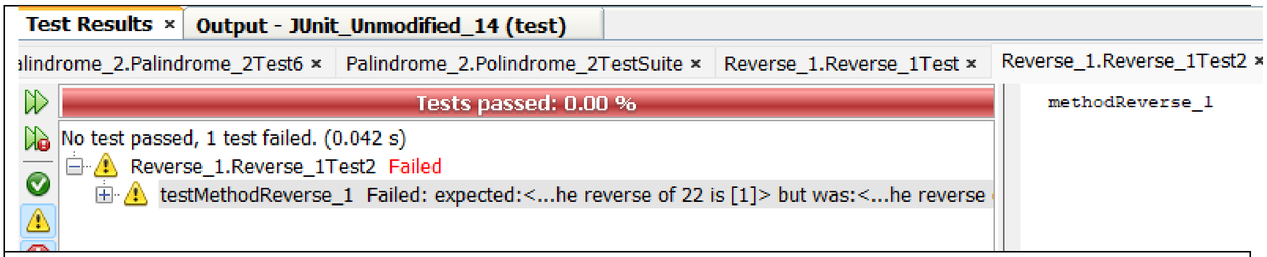
j. Input: 22

- **Snippet of test case**



```
    @Test
    public void testMethodReverse_1() {
        System.out.println("methodReverse_1");
        int n = 22;
        Reverse_1 instance = new Reverse_1();
        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**

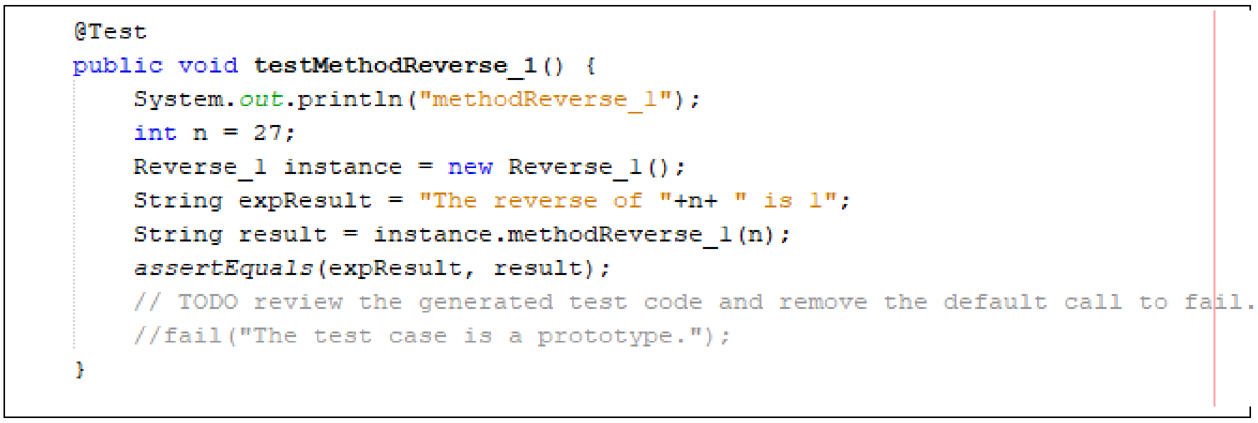


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
alindrome_2.Palindrome_2Test6 ×   Palindrome_2.Polindrome_2TestSuite ×   Reverse_1.Reverse_1Test ×   Reverse_1.Reverse_1Test2 ×
Tests passed: 0.00 %
No test passed, 1 test failed. (0.042 s)
  Reverse_1.Reverse_1Test2  Failed
    testMethodReverse_1  Failed: expected:<...he reverse of 22 is [1]> but was:<...he reverse
methodReverse_1
```

Penjelasan : Test ini tidak berhasil karena ada perbedaan antara hasil yang diharapkan dan hasil actual dari metode ‘methodReverse_1” karena exxresult is 1 seharusnya 22

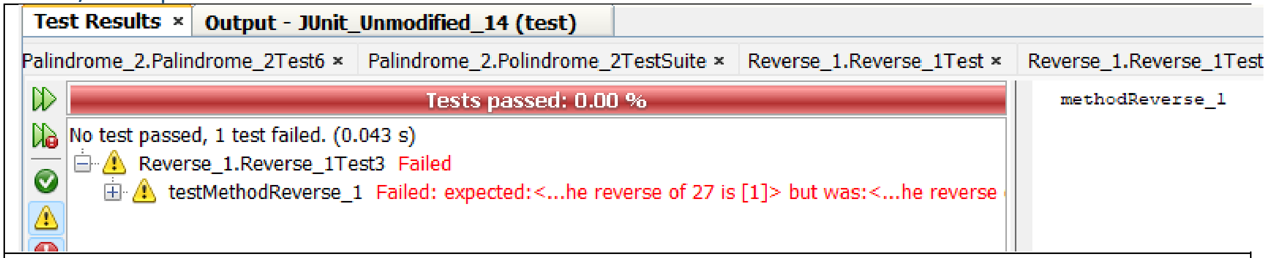
k. Input: 27

- **Snippet of test case**



```
    @Test
    public void testMethodReverse_1() {
        System.out.println("methodReverse_1");
        int n = 27;
        Reverse_1 instance = new Reverse_1();
        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**

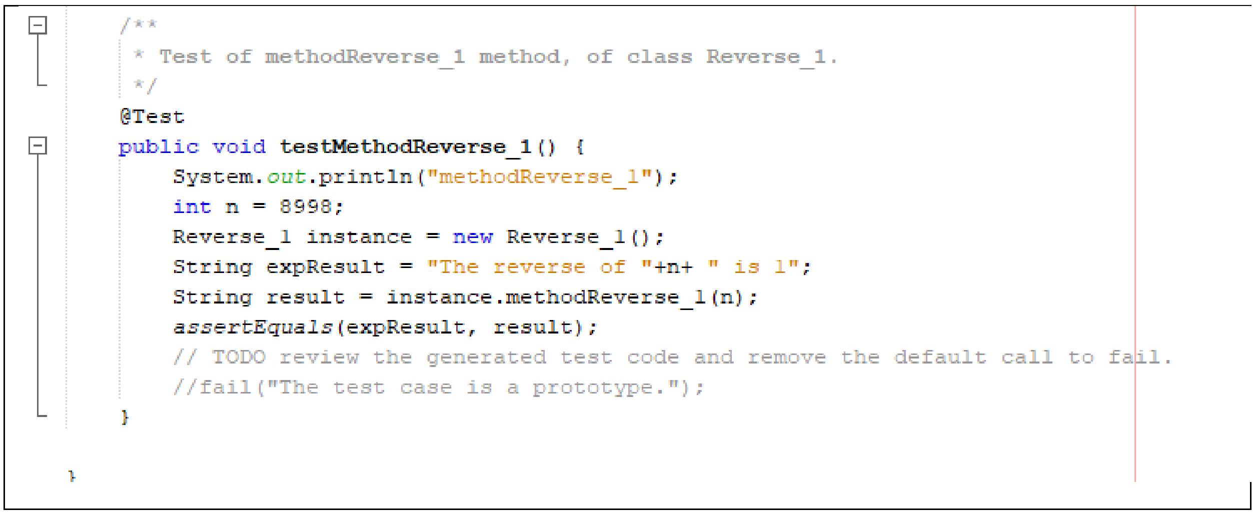


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
Palindrome_2.Palindrome_2Test6 ×   Palindrome_2.Polindrome_2TestSuite ×   Reverse_1.Reverse_1Test ×   Reverse_1.Reverse_1Test
Tests passed: 0.00 %                                                                   methodReverse_1
No test passed, 1 test failed. (0.043 s)
  Reverse_1.Reverse_1Test3  Failed
    testMethodReverse_1  Failed: expected:<...he reverse of 27 is [1]> but was:<...he reverse
```

Penjelasan : Test ini tidak berhasil karena ada perbedaan antara hasil yang diharapkan dan hasil actual dari metode 'methodReverse_1" karena exxresult is 1 seharusnya 27.

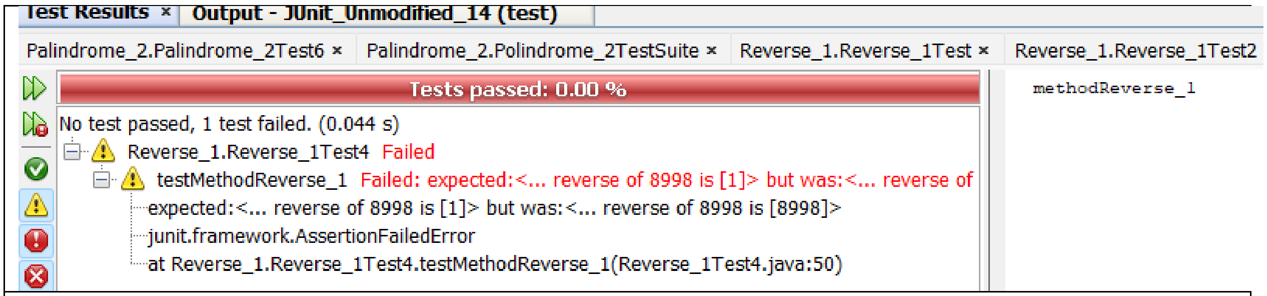
l. **Input: 8998**

➢ **Snippet of test case**



```
    /**
     * Test of methodReverse_1 method, of class Reverse_1.
     */
    @Test
    public void testMethodReverse_1() {
        System.out.println("methodReverse_1");
        int n = 8998;
        Reverse_1 instance = new Reverse_1();
        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**

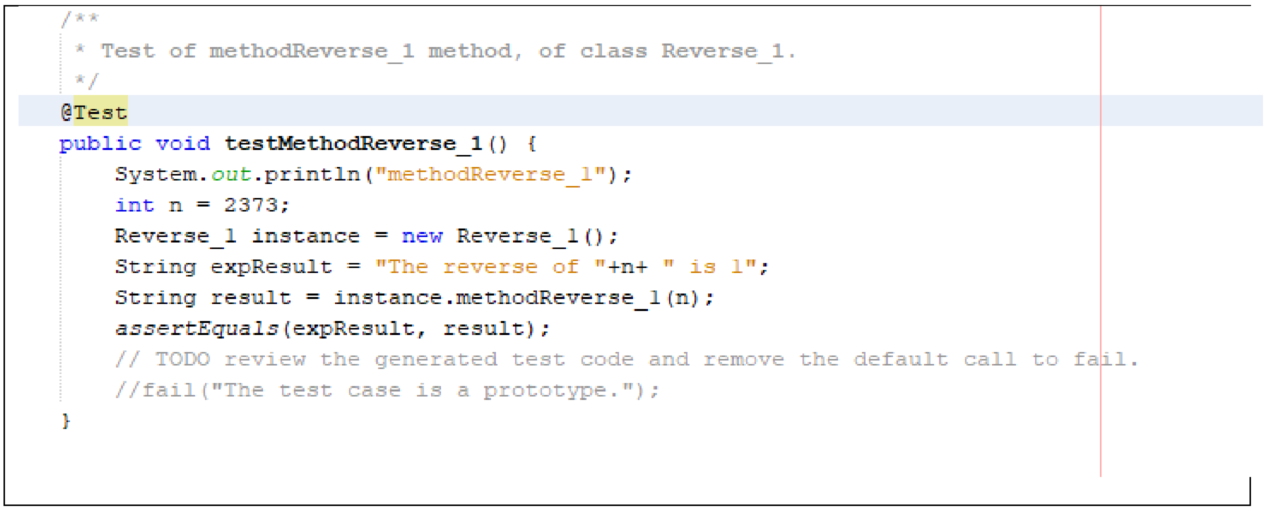


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
Palindrome_2.Palindrome_2Test6 ×   Palindrome_2.Polindrome_2TestSuite ×   Reverse_1.Reverse_1Test ×   Reverse_1.Reverse_1Test2
Tests passed: 0.00 %                                                                   methodReverse_1
No test passed, 1 test failed. (0.044 s)
  Reverse_1.Reverse_1Test4  Failed
    testMethodReverse_1  Failed: expected:<... reverse of 8998 is [1]> but was:<... reverse of
      expected:<... reverse of 8998 is [1]> but was:<... reverse of 8998 is [8998]>
      junit.framework.AssertionFailedError
      at Reverse_1.Reverse_1Test4.testMethodReverse_1(Reverse_1Test4.java:50)
```

Penjelasan : Test ini tidak berhasil karena ada perbedaan antara hasil yang diharapkan dan hasil actual dari metode 'methodReverse_1" karena exxresult is 1 seharusnya 8998.

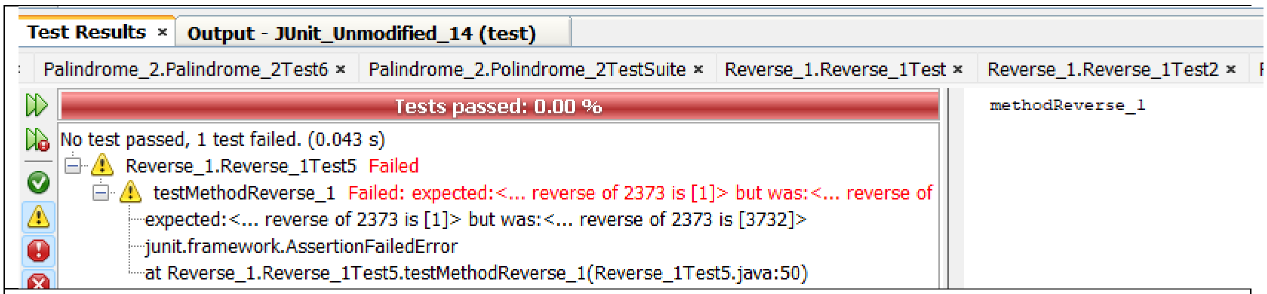
m. Input: 2373

➢ **Snippet of test case**



```
/**
 * Test of methodReverse_1 method, of class Reverse_1.
 */
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 2373;
    Reverse_1 instance = new Reverse_1();
    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**

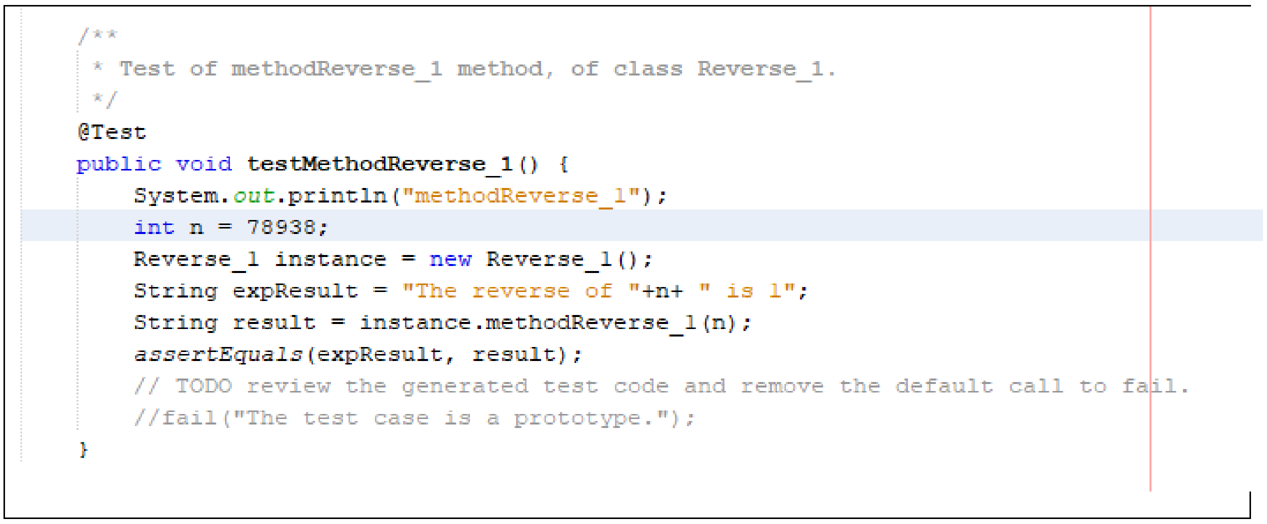


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
Palindrome_2.Palindrome_2Test6 ×   Palindrome_2.Polindrome_2TestSuite ×   Reverse_1.Reverse_1Test ×   Reverse_1.Reverse_1Test2 ×
Tests passed: 0.00 %                                                    methodReverse_1
No test passed, 1 test failed. (0.043 s)
  Reverse_1.Reverse_1Test5  Failed
    testMethodReverse_1  Failed: expected:<... reverse of 2373 is [1]> but was:<... reverse of
      expected:<... reverse of 2373 is [1]> but was:<... reverse of 2373 is [3732]>
      junit.framework.AssertionFailedError
      at Reverse_1.Reverse_1Test5.testMethodReverse_1(Reverse_1Test5.java:50)
```

Penjelasan : Test ini tidak berhasil karena ada perbedaan antara hasil yang diharapkan dan hasil actual dari metode ‘methodReverse_1” karena exxresult is 1 seharusnya 2373.

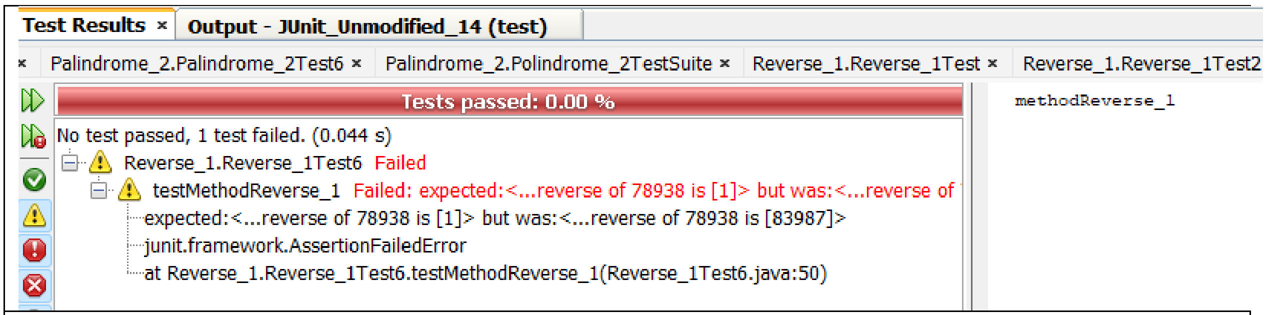
n. Input: 78938

➢ **Snippet of test case**



```
/**
 * Test of methodReverse_1 method, of class Reverse_1.
 */
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 78938;
    Reverse_1 instance = new Reverse_1();
    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**



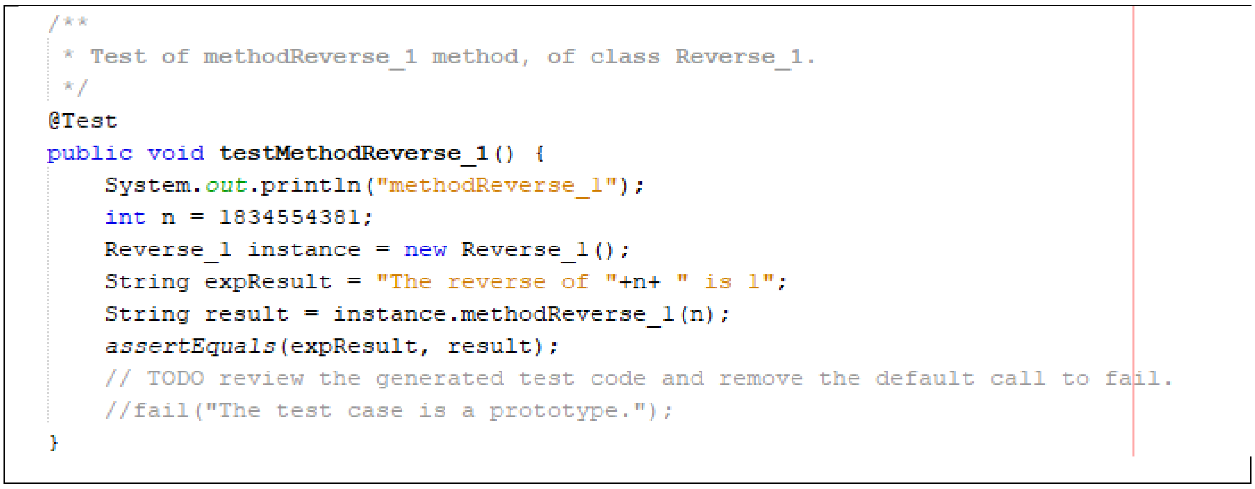
Test Results
Output - JUnit_Unmodified_14 (test)
Palindrome_2.Palindrome_2Test6
Palindrome_2.Polindrome_2TestSuite
Reverse_1.Reverse_1Test
Reverse_1.Reverse_1Test2
Tests passed: 0.00 %
methodReverse_1

```
No test passed, 1 test failed. (0.044 s)
  Reverse_1.Reverse_1Test6  Failed
    testMethodReverse_1  Failed: expected:<...reverse of 78938 is [1]> but was:<...reverse of
      expected:<...reverse of 78938 is [1]> but was:<...reverse of 78938 is [83987]>
      junit.framework.AssertionFailedError
      at Reverse_1.Reverse_1Test6.testMethodReverse_1(Reverse_1Test6.java:50)
```

Penjelasan : Test ini tidak berhasil karena ada perbedaan antara hasil yang diharapkan dan hasil actual dari metode ‘methodReverse_1” karena exxresult is 1 seharusnya 78938.

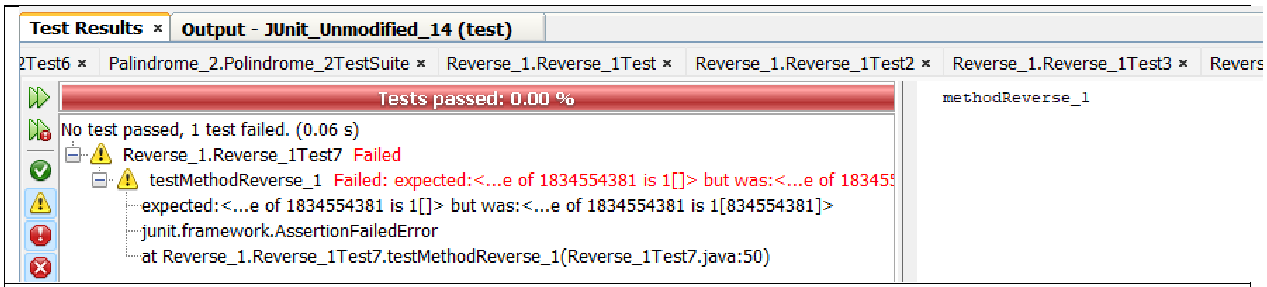
o. Input: 1834554381

- **Snippet of test case**



```
/**
 * Test of methodReverse_1 method, of class Reverse_1.
 */
@Test
public void testMethodReverse_1() {
    System.out.println("methodReverse_1");
    int n = 1834554381;
    Reverse_1 instance = new Reverse_1();
    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**

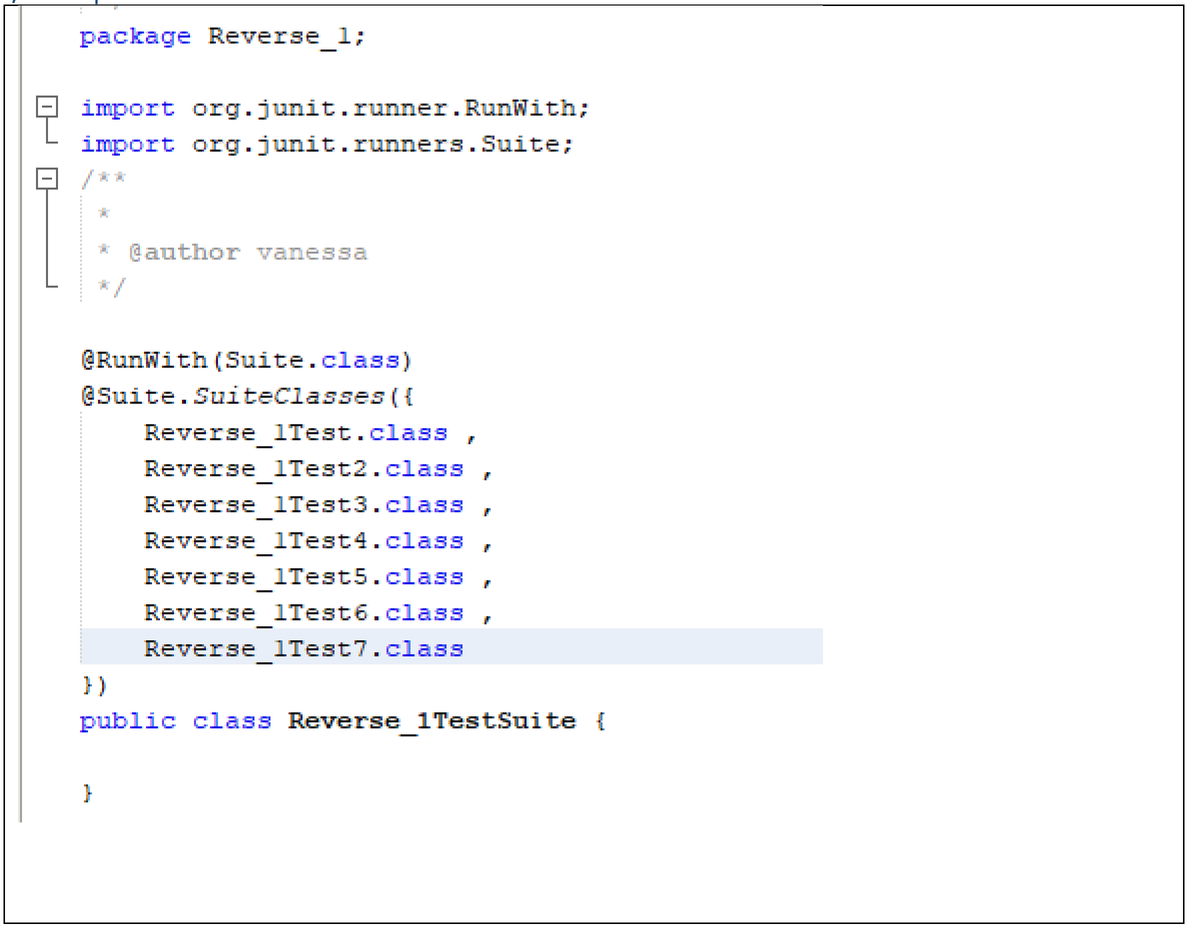


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
2Test6 ×  Palindrome_2.Polindrome_2TestSuite ×  Reverse_1.Reverse_1Test ×  Reverse_1.Reverse_1Test2 ×  Reverse_1.Reverse_1Test3 ×  Revers
Tests passed: 0.00 %                                                methodReverse_1
No test passed, 1 test failed. (0.06 s)
Reverse_1.Reverse_1Test7  Failed
   testMethodReverse_1  Failed: expected:<...e of 1834554381 is 1[]> but was:<...e of 183455
      expected:<...e of 1834554381 is 1[]> but was:<...e of 1834554381 is 1[834554381]>
      junit.framework.AssertionFailedError
      at Reverse_1.Reverse_1Test7.testMethodReverse_1(Reverse_1Test7.java:50)
```

Penjelasan : Test ini tidak berhasil karena ada perbedaan antara hasil yang diharapkan dan hasil actual dari metode ‘methodReverse_1” karena exxresult is 1 seharusnya 2834554381.

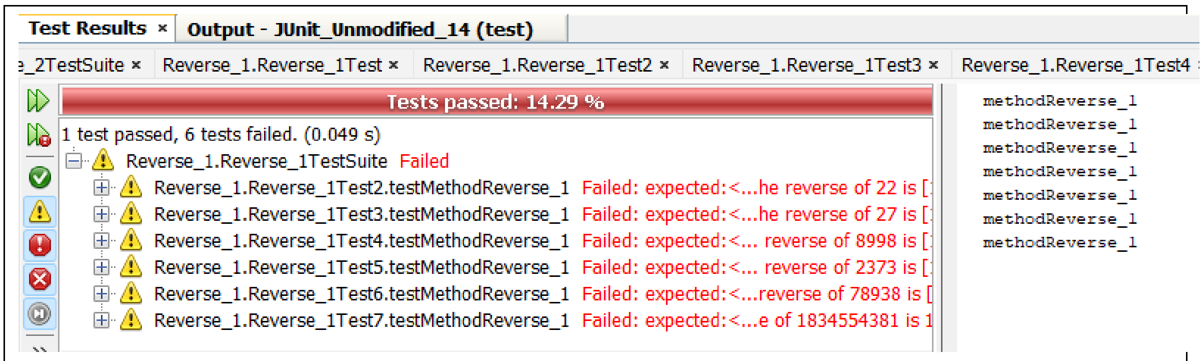
p. TestSuite

- **Snippet of test case**



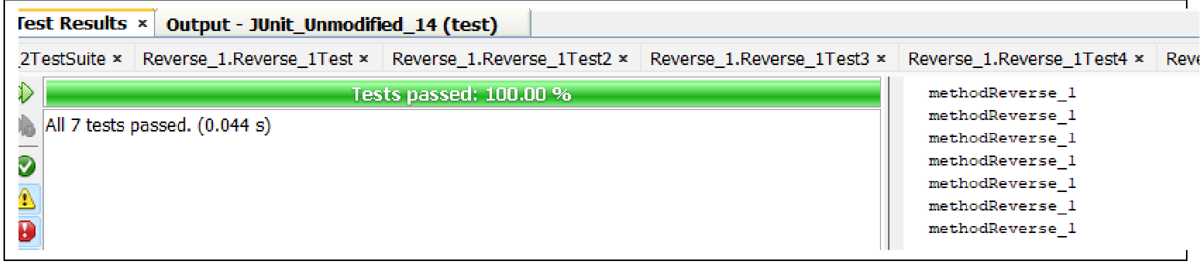
```
package Reverse_1;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;
/**
 *
 * @author vanessa
 */

@RunWith(Suite.class)
@Suite.SuiteClasses({
    Reverse_1Test.class ,
    Reverse_1Test2.class ,
    Reverse_1Test3.class ,
    Reverse_1Test4.class ,
    Reverse_1Test5.class ,
    Reverse_1Test6.class ,
    Reverse_1Test7.class
})
public class Reverse_1TestSuite {

}
```

➤ **Snippet of results**



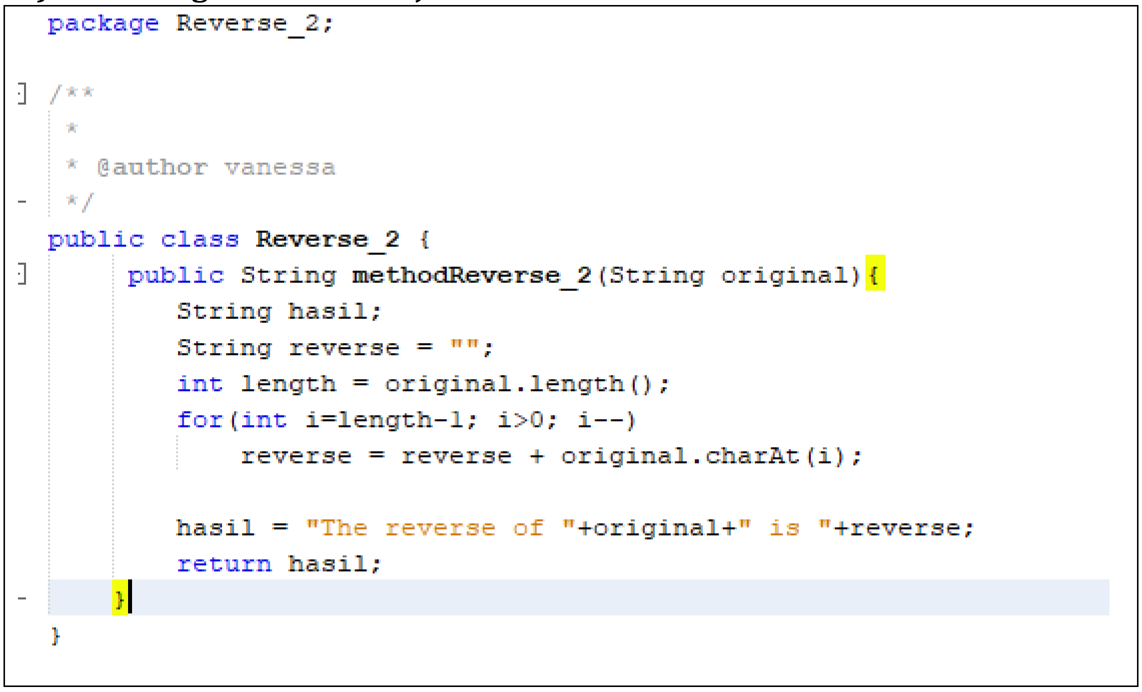
```
Test Results ×   Output - JUnit_Unmodified_14 (test)
e_2TestSuite ×  Reverse_1.Reverse_1Test ×  Reverse_1.Reverse_1Test2 ×  Reverse_1.Reverse_1Test3 ×  Reverse_1.Reverse_1Test4
                         Tests passed: 14.29 %
1 test passed, 6 tests failed. (0.049 s)
  Reverse_1.Reverse_1TestSuite  Failed
    Reverse_1.Reverse_1Test2.testMethodReverse_1  Failed: expected:<...he reverse of 22 is [
    Reverse_1.Reverse_1Test3.testMethodReverse_1  Failed: expected:<...he reverse of 27 is [
    Reverse_1.Reverse_1Test4.testMethodReverse_1  Failed: expected:<... reverse of 8998 is [
    Reverse_1.Reverse_1Test5.testMethodReverse_1  Failed: expected:<... reverse of 2373 is [
    Reverse_1.Reverse_1Test6.testMethodReverse_1  Failed: expected:<...reverse of 78938 is [
    Reverse_1.Reverse_1Test7.testMethodReverse_1  Failed: expected:<...e of 1834554381 is 1

methodReverse_1
methodReverse_1
methodReverse_1
methodReverse_1
methodReverse_1
methodReverse_1
methodReverse_1
```

Penjelasan : Beberapa test tidak berhasil dikarenakan terdapat ada yang tidak sesuai, namun dalam eror sudah dimodif untuk kesesuain maka hasilnya akan menjadi seperti tapi hasilnya akan seperti dibawa ini



```
Test Results ×   Output - JUnit_Unmodified_14 (test)
2TestSuite ×  Reverse_1.Reverse_1Test ×  Reverse_1.Reverse_1Test2 ×  Reverse_1.Reverse_1Test3 ×  Reverse_1.Reverse_1Test4 ×  Reve
                         Tests passed: 100.00 %
All 7 tests passed. (0.044 s)

methodReverse_1
methodReverse_1
methodReverse_1
methodReverse_1
methodReverse_1
methodReverse_1
methodReverse_1
```

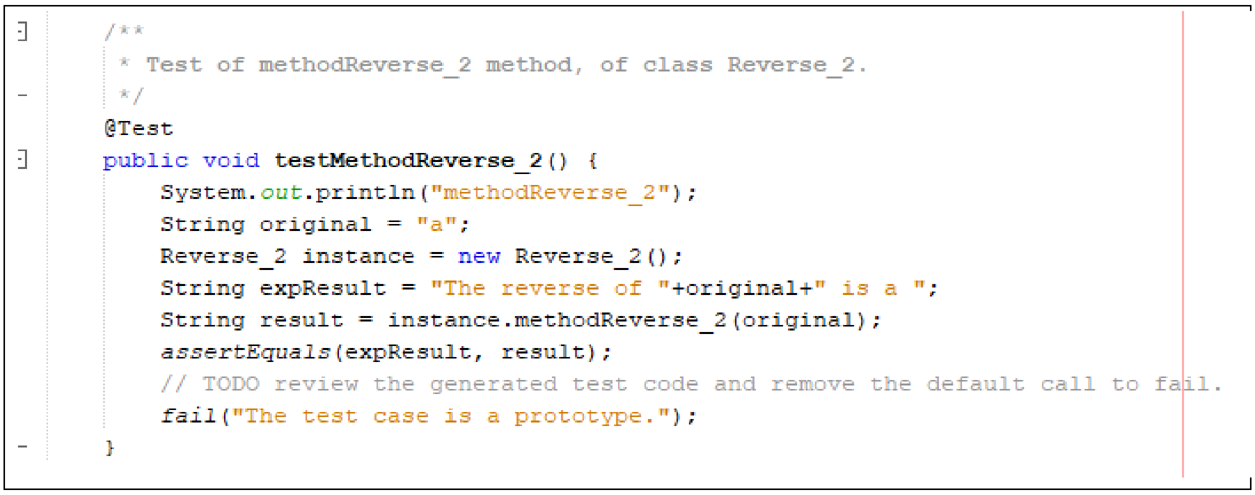
4) Testing for Reverse_2.java



```
package Reverse_2;

/**
 *
 * @author vanessa
 */
public class Reverse_2 {
    public String methodReverse_2(String original){
        String hasil;
        String reverse = "";
        int length = original.length();
        for(int i=length-1; i>0; i--)
            reverse = reverse + original.charAt(i);

        hasil = "The reverse of "+original+" is "+reverse;
        return hasil;
    }
}
```

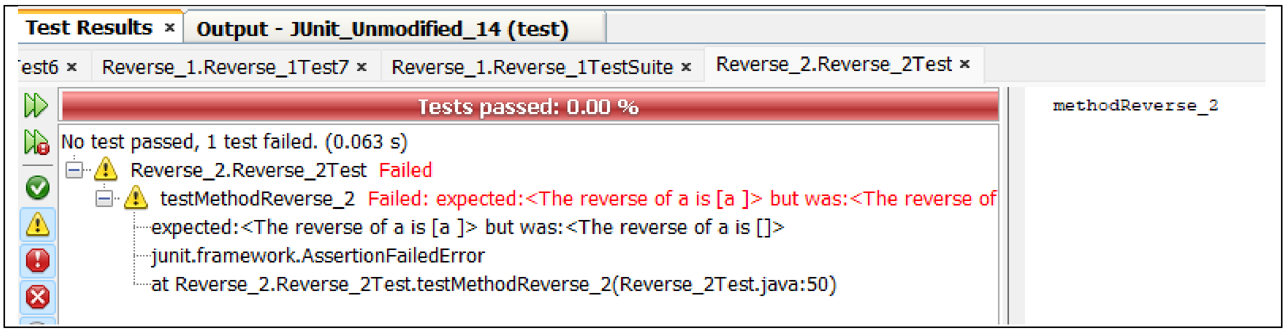
a. Input: a

➢ **Snippet of test case**



```
    /**
     * Test of methodReverse_2 method, of class Reverse_2.
     */
    @Test
    public void testMethodReverse_2() {
        System.out.println("methodReverse_2");
        String original = "a";
        Reverse_2 instance = new Reverse_2();
        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**

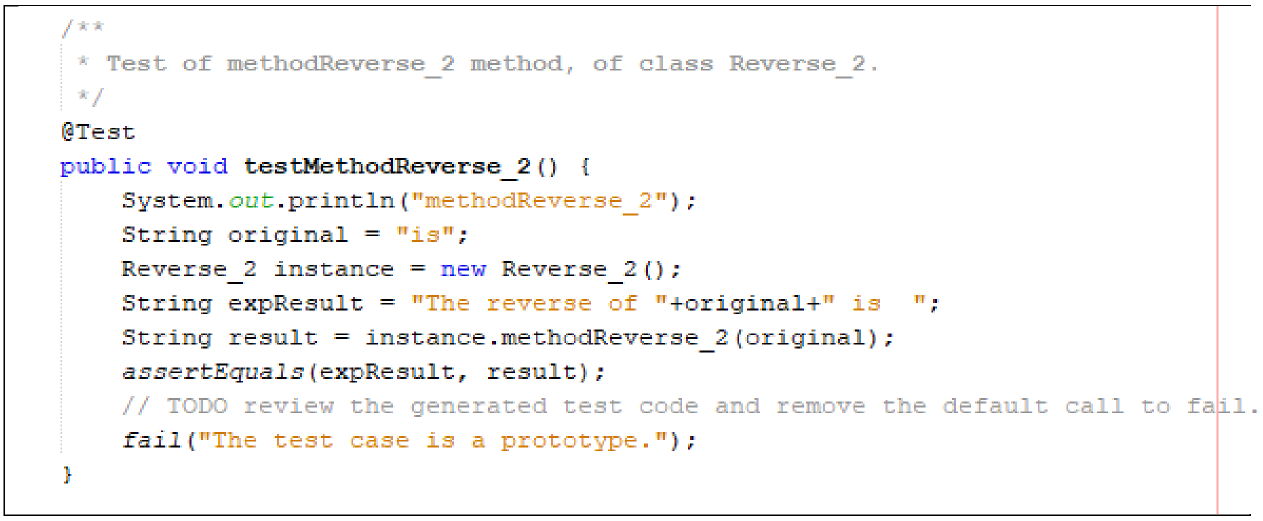


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
'est6 ×   Reverse_1.Reverse_1Test7 ×   Reverse_1.Reverse_1TestSuite ×   Reverse_2.Reverse_2Test ×

Tests passed: 0.00 %                                                    methodReverse_2
No test passed, 1 test failed. (0.063 s)
⊟ Reverse_2.Reverse_2Test  Failed
   ⊟ testMethodReverse_2  Failed: expected:<The reverse of a is [a ]> but was:<The reverse of
        expected:<The reverse of a is [a ]> but was:<The reverse of a is []>
        junit.framework.AssertionFailedError
        at Reverse_2.Reverse_2Test.testMethodReverse_2(Reverse_2Test.java:50)
```

Penjelasan : Test testMethodReverse_2 mengalami error karena ada kesalahan dalam penghitungan indeks dalam perulangan untuk membalikkan string pada metode methodReverse_2 karena hasil yang diharapkan (expected) dan hasil aktual (actual) tidak sesuai. Ini disebabkan oleh sebuah spasi yang terdapat di hasil aktual, yaitu "a " (a dengan spasi di belakangnya), sementara hasil yang diharapkan adalah "a" tanpa spasi.

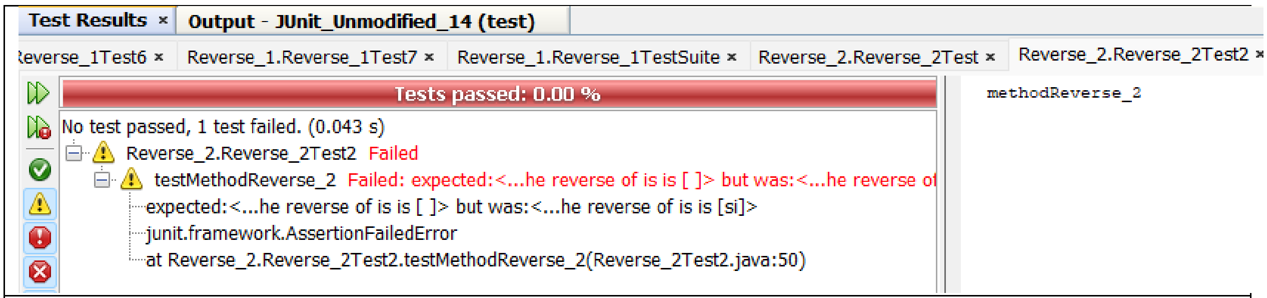
b. Input: is

➢ **Snippet of test case**



```
/**
 * Test of methodReverse_2 method, of class Reverse_2.
 */
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "is";
    Reverse_2 instance = new Reverse_2();
    String expResult = "The reverse of "+original+" 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**

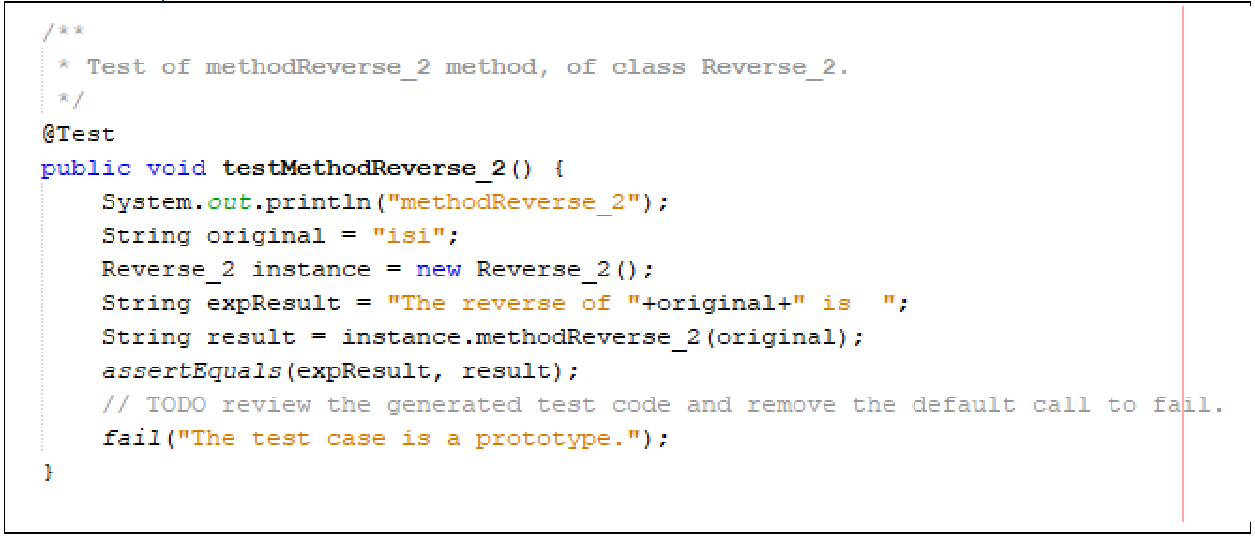


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
everse_1Test6 ×  Reverse_1.Reverse_1Test7 ×  Reverse_1.Reverse_1TestSuite ×  Reverse_2.Reverse_2Test ×  Reverse_2.Reverse_2Test2 ×

Tests passed: 0.00 %                                                     methodReverse_2
No test passed, 1 test failed. (0.043 s)
Reverse_2.Reverse_2Test2  Failed
  testMethodReverse_2  Failed: expected:<...he reverse of is is [ ]> but was:<...he reverse of
    expected:<...he reverse of is is [ ]> but was:<...he reverse of is is [si]>
    junit.framework.AssertionFailedError
    at Reverse_2.Reverse_2Test2.testMethodReverse_2(Reverse_2Test2.java:50)
```

Penjelasan : Test ini mengalami error karena hasil yang diharapkan (expected result) dan hasil aktual (actual result) tidak sesuai. Perhatikan perbandingan hasilnya Expected result: "The reverse of a is "dan Actual result: "The reverse of a is "
Dalam pesan error yang di berikan, kedua hasil tersebut terlihat sama. Namun, perbedaan yang sebenarnya terjadi adalah spasi di belakang string "The reverse of a is " pada hasil yang diharapkan (expected result). Spasi tersebut tidak terdapat pada hasil aktual (actual result).

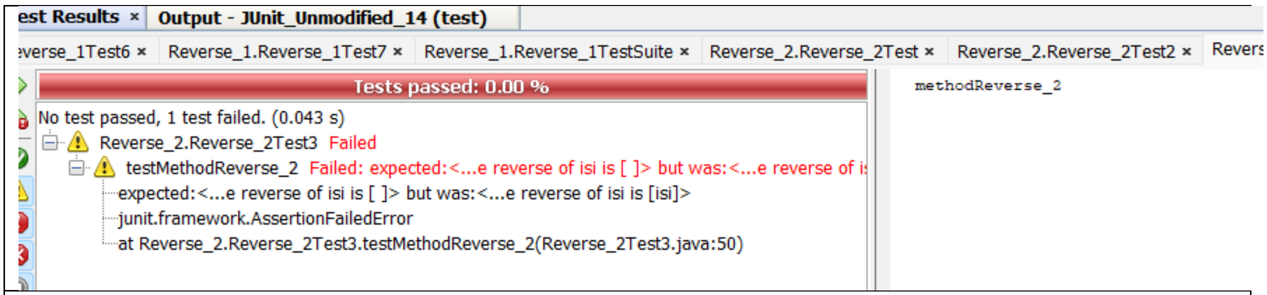
c. Input: isi

➢ **Snippet of test case**



```
/**
 * Test of methodReverse_2 method, of class Reverse_2.
 */
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "isi";
    Reverse_2 instance = new Reverse_2();
    String expResult = "The reverse of "+original+" 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**

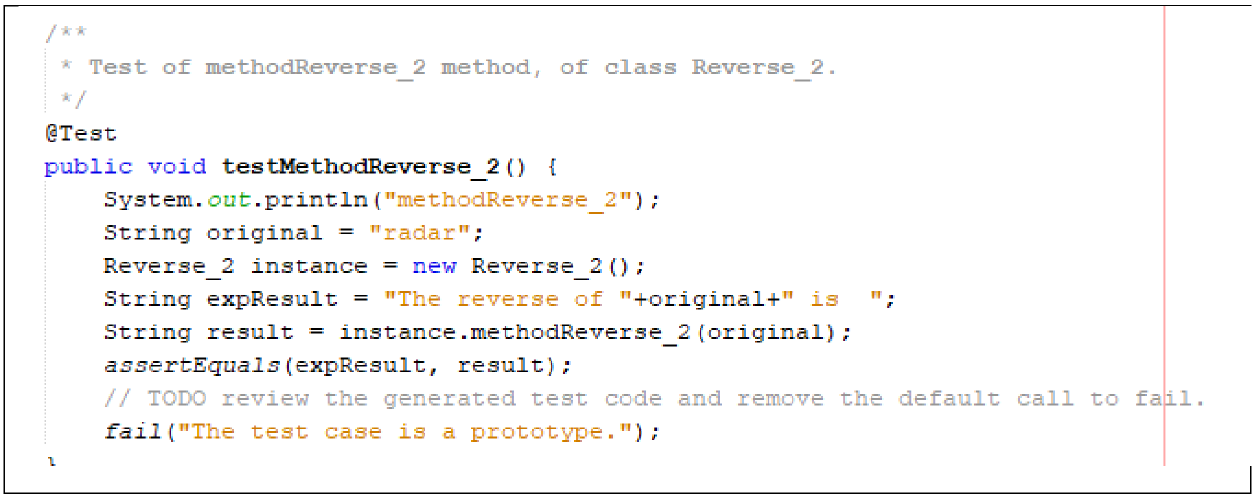


```
est Results × | Output - JUnit_Unmodified_14 (test)
everse_1Test6 × | Reverse_1.Reverse_1Test7 × | Reverse_1.Reverse_1TestSuite × | Reverse_2.Reverse_2Test × | Reverse_2.Reverse_2Test2 × | Revers
Tests passed: 0.00 %                                                        methodReverse_2
No test passed, 1 test failed. (0.043 s)
  Reverse_2.Reverse_2Test3 Failed
    testMethodReverse_2 Failed: expected:<...e reverse of isi is [ ]> but was:<...e reverse of i
      expected:<...e reverse of isi is [ ]> but was:<...e reverse of isi is [isi]>
      junit.framework.AssertionFailedError
      at Reverse_2.Reverse_2Test3.testMethodReverse_2(Reverse_2Test3.java:50)
```

Penjelasan : Test ini mengalami error karena hasil yang diharapkan (expected result) dan hasil aktual (actual result) tidak sesuai. Perhatikan perbandingan hasilnya Expected result: "The reverse of a is "dan Actual result: "The reverse of a is "
Dalam pesan error yang di berikan, kedua hasil tersebut terlihat sama. Namun, perbedaan yang sebenarnya terjadi adalah spasi di belakang string "The reverse of a is " pada hasil yang diharapkan (expected result). Spasi tersebut tidak terdapat pada hasil aktual (actual result).

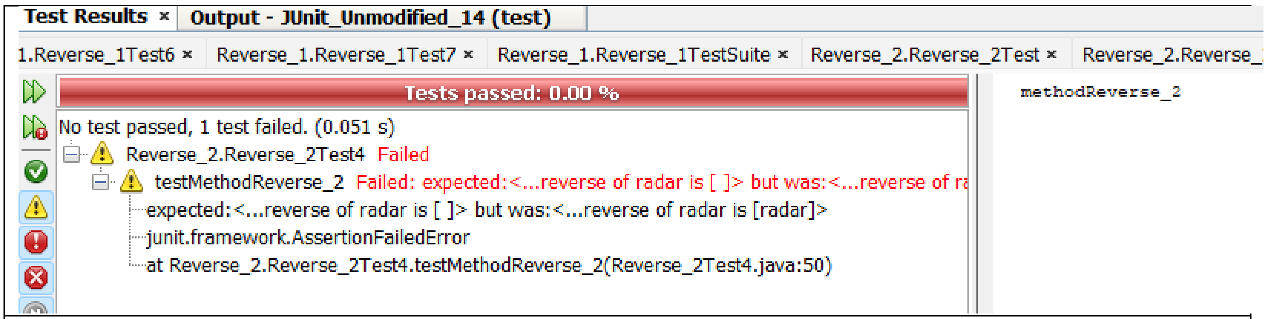
d. Input: radar

➢ **Snippet of test case**



```
/**
 * Test of methodReverse_2 method, of class Reverse_2.
 */
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "radar";
    Reverse_2 instance = new Reverse_2();
    String expResult = "The reverse of "+original+" 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**



```
Test Results ×  Output - JUnit_Unmodified_14 (test)
1.Reverse_1Test6 ×  Reverse_1.Reverse_1Test7 ×  Reverse_1.Reverse_1TestSuite ×  Reverse_2.Reverse_2Test ×  Reverse_2.Reverse_
Tests passed: 0.00 %
No test passed, 1 test failed. (0.051 s)
  Reverse_2.Reverse_2Test4  Failed
    testMethodReverse_2  Failed: expected:<...reverse of radar is [ ]> but was:<...reverse of ra
      expected:<...reverse of radar is [ ]> but was:<...reverse of radar is [radar]>
      junit.framework.AssertionFailedError
      at Reverse_2.Reverse_2Test4.testMethodReverse_2(Reverse_2Test4.java:50)
```

```
methodReverse_2
```

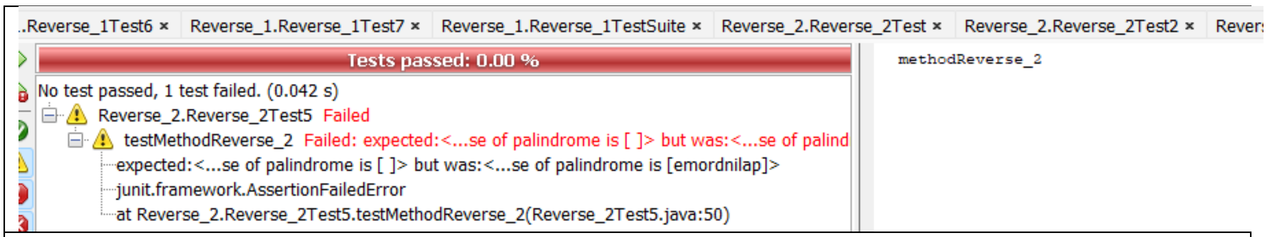
Penjelasan : Test ini mengalami error karena hasil yang diharapkan (expected result) dan hasil aktual (actual result) tidak sesuai. Perhatikan perbandingan hasilnya Expected result: "The reverse of a is "dan Actual result: "The reverse of a is "
Dalam pesan error yang di berikan, kedua hasil tersebut terlihat sama. Namun, perbedaan yang sebenarnya terjadi adalah spasi di belakang string "The reverse of a is " pada hasil yang diharapkan (expected result). Spasi tersebut tidak terdapat pada hasil aktual (actual result).

e. Input: palindrome

➢ **Snippet of test case**

```
/**
 * Test of methodReverse_2 method, of class Reverse_2.
 */
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "palindrome";
    Reverse_2 instance = new Reverse_2();
    String expResult = "The reverse of "+original+" 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**



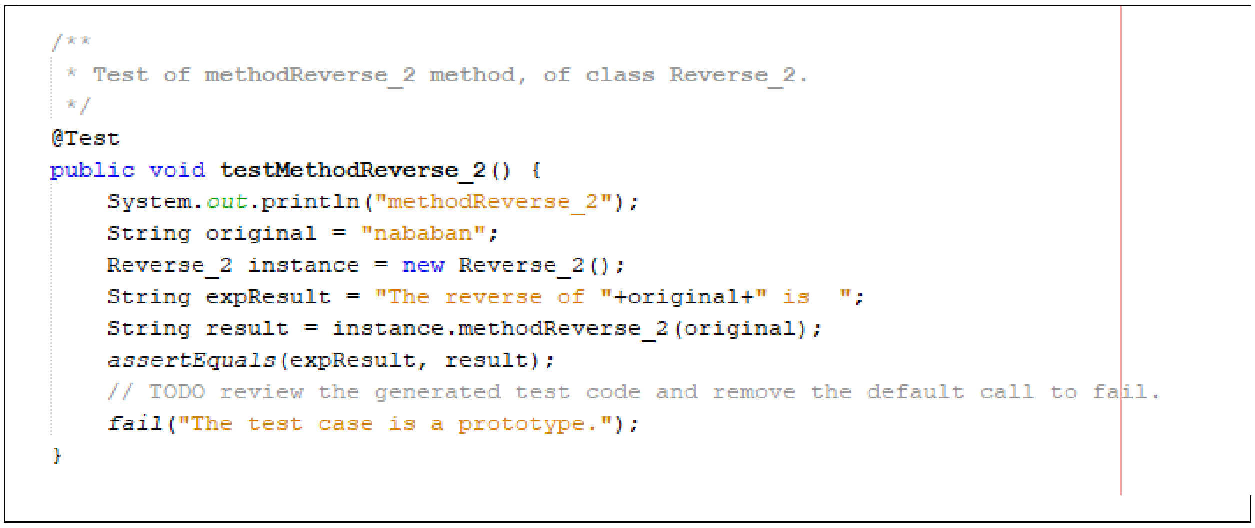
```
.Reverse_1Test6 ×  Reverse_1.Reverse_1Test7 ×  Reverse_1.Reverse_1TestSuite ×  Reverse_2.Reverse_2Test ×  Reverse_2.Reverse_2Test2 ×  Rever
Tests passed: 0.00 %
No test passed, 1 test failed. (0.042 s)
  Reverse_2.Reverse_2Test5  Failed
    testMethodReverse_2  Failed: expected:<...se of palindrome is [ ]> but was:<...se of palind
      expected:<...se of palindrome is [ ]> but was:<...se of palindrome is [emordnilap]>
      junit.framework.AssertionFailedError
      at Reverse_2.Reverse_2Test5.testMethodReverse_2(Reverse_2Test5.java:50)
```

```
methodReverse_2
```

Penjelasan : Test ini mengalami error karena hasil yang diharapkan (expected result) dan hasil aktual (actual result) tidak sesuai. Perhatikan perbandingan hasilnya Expected result: "The reverse of a is "dan Actual result: "The reverse of a is "
Dalam pesan error yang di berikan, kedua hasil tersebut terlihat sama. Namun, perbedaan yang sebenarnya terjadi adalah spasi di belakang string "The reverse of a is " pada hasil yang diharapkan (expected result). Spasi tersebut tidak terdapat pada hasil aktual (actual result).

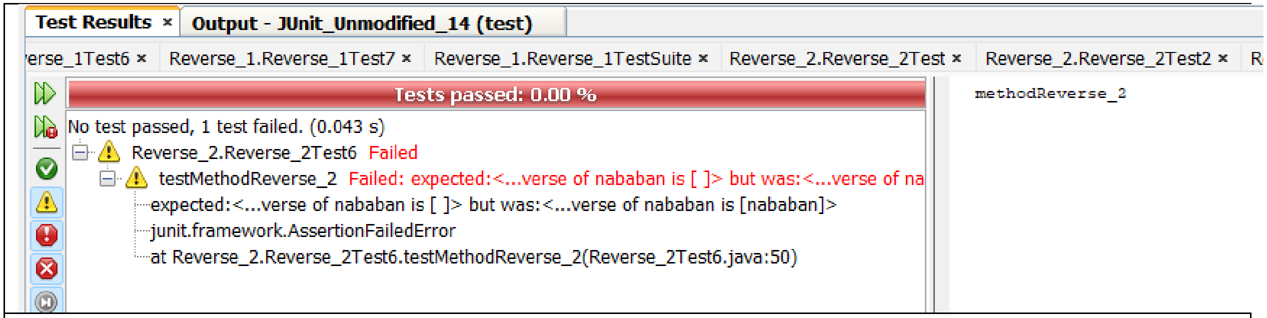
f. Input: nababan

➢ **Snippet of test case**



```
/**
 * Test of methodReverse_2 method, of class Reverse_2.
 */
@Test
public void testMethodReverse_2() {
    System.out.println("methodReverse_2");
    String original = "nababan";
    Reverse_2 instance = new Reverse_2();
    String expResult = "The reverse of "+original+" 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**

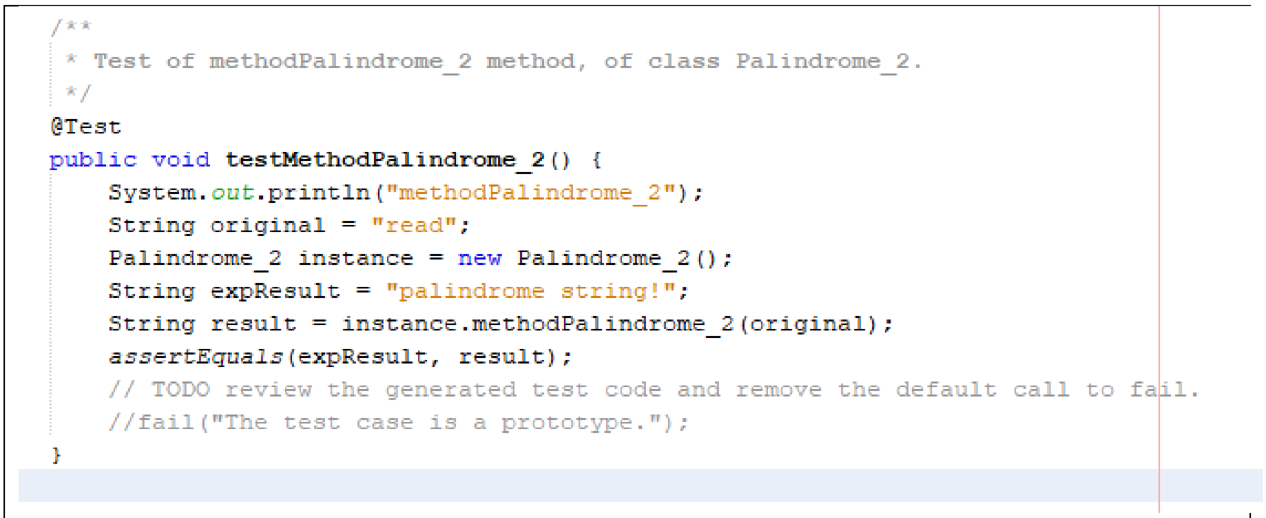


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
erse_1Test6 ×  Reverse_1.Reverse_1Test7 ×  Reverse_1.Reverse_1TestSuite ×  Reverse_2.Reverse_2Test ×  Reverse_2.Reverse_2Test2 ×  R
Tests passed: 0.00 %                                                          methodReverse_2
No test passed, 1 test failed. (0.043 s)
  Reverse_2.Reverse_2Test6  Failed
    testMethodReverse_2  Failed: expected:<...verse of nababan is [ ]> but was:<...verse of na
      expected:<...verse of nababan is [ ]> but was:<...verse of nababan is [nababan]>
      junit.framework.AssertionFailedError
      at Reverse_2.Reverse_2Test6.testMethodReverse_2(Reverse_2Test6.java:50)
```

Penjelasan : Test ini mengalami error karena hasil yang diharapkan (expected result) dan hasil aktual (actual result) tidak sesuai. Perhatikan perbandingan hasilnya Expected result: "The reverse of a is "dan Actual result: "The reverse of a is "
Dalam pesan error yang di berikan, kedua hasil tersebut terlihat sama. Namun, perbedaan yang sebenarnya terjadi adalah spasi di belakang string "The reverse of a is " pada hasil yang diharapkan (expected result). Spasi tersebut tidak terdapat pada hasil aktual (actual result).

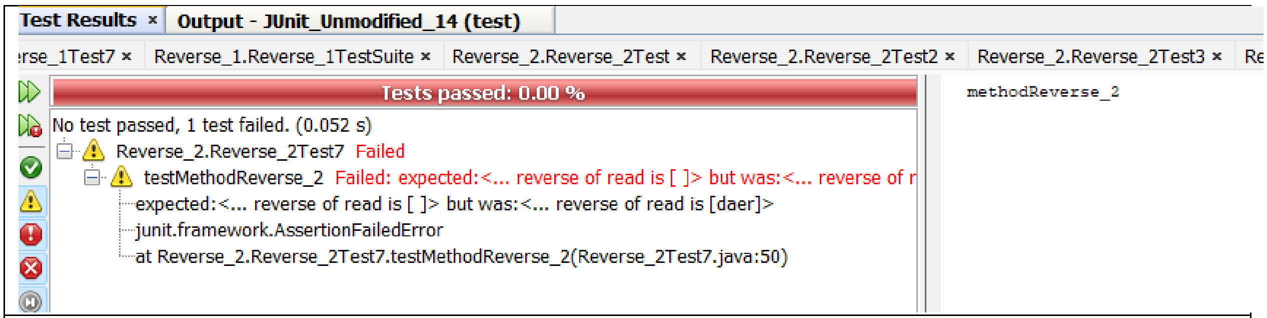
g. Input: read

➢ **Snippet of test case**



```
/**
 * Test of methodPalindrome_2 method, of class Palindrome_2.
 */
@Test
public void testMethodPalindrome_2() {
    System.out.println("methodPalindrome_2");
    String original = "read";
    Palindrome_2 instance = new Palindrome_2();
    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**



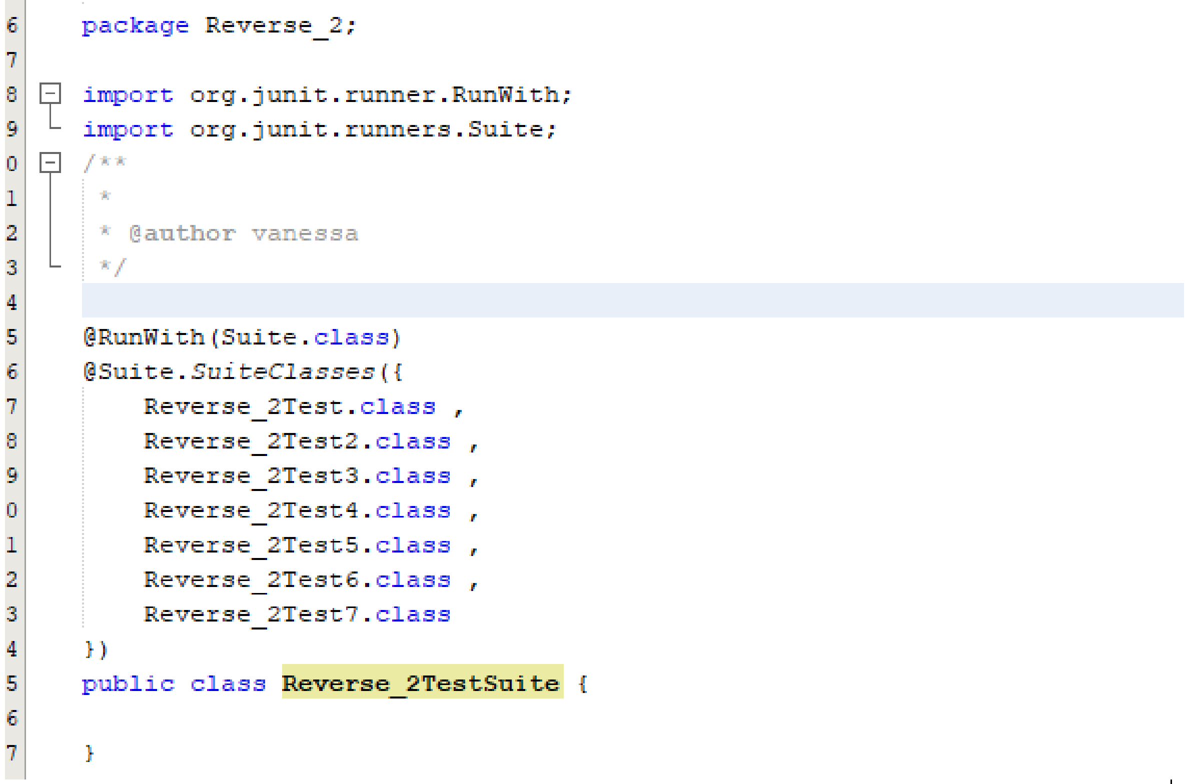
```
Tests passed: 0.00 %
No test passed, 1 test failed. (0.052 s)
Reverse_2.Reverse_2Test7  Failed
  testMethodReverse_2  Failed: expected:<... reverse of read is [ ]> but was:<... reverse of r
    expected:<... reverse of read is [ ]> but was:<... reverse of read is [daer]>
    junit.framework.AssertionFailedError
    at Reverse_2.Reverse_2Test7.testMethodReverse_2(Reverse_2Test7.java:50)
```

```
methodReverse_2
```

Penjelasan : Test ini mengalami error karena hasil yang diharapkan (expected result) dan hasil aktual (actual result) tidak sesuai. Perhatikan perbandingan hasilnya Expected result: "The reverse of a is "dan Actual result: "The reverse of a is "
Dalam pesan error yang di berikan, kedua hasil tersebut terlihat sama. Namun, perbedaan yang sebenarnya terjadi adalah spasi di belakang string "The reverse of a is " pada hasil yang diharapkan (expected result). Spasi tersebut tidak terdapat pada hasil aktual (actual result).

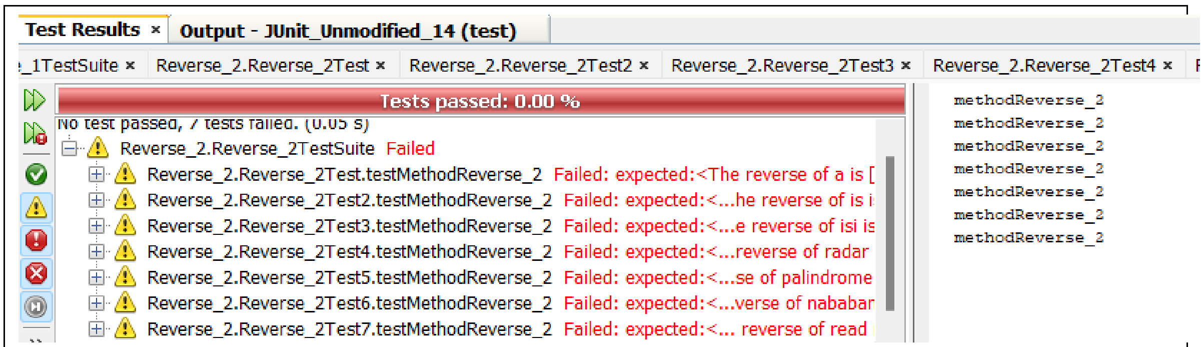
h. TestSuite

➢ **Snippet of test case**



```
package Reverse_2;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;
/**
 *
 * @author vanessa
 */

@RunWith(Suite.class)
@Suite.SuiteClasses({
    Reverse_2Test.class ,
    Reverse_2Test2.class ,
    Reverse_2Test3.class ,
    Reverse_2Test4.class ,
    Reverse_2Test5.class ,
    Reverse_2Test6.class ,
    Reverse_2Test7.class
})
public class Reverse_2TestSuite {

}
```

➢ **Snippet of results**

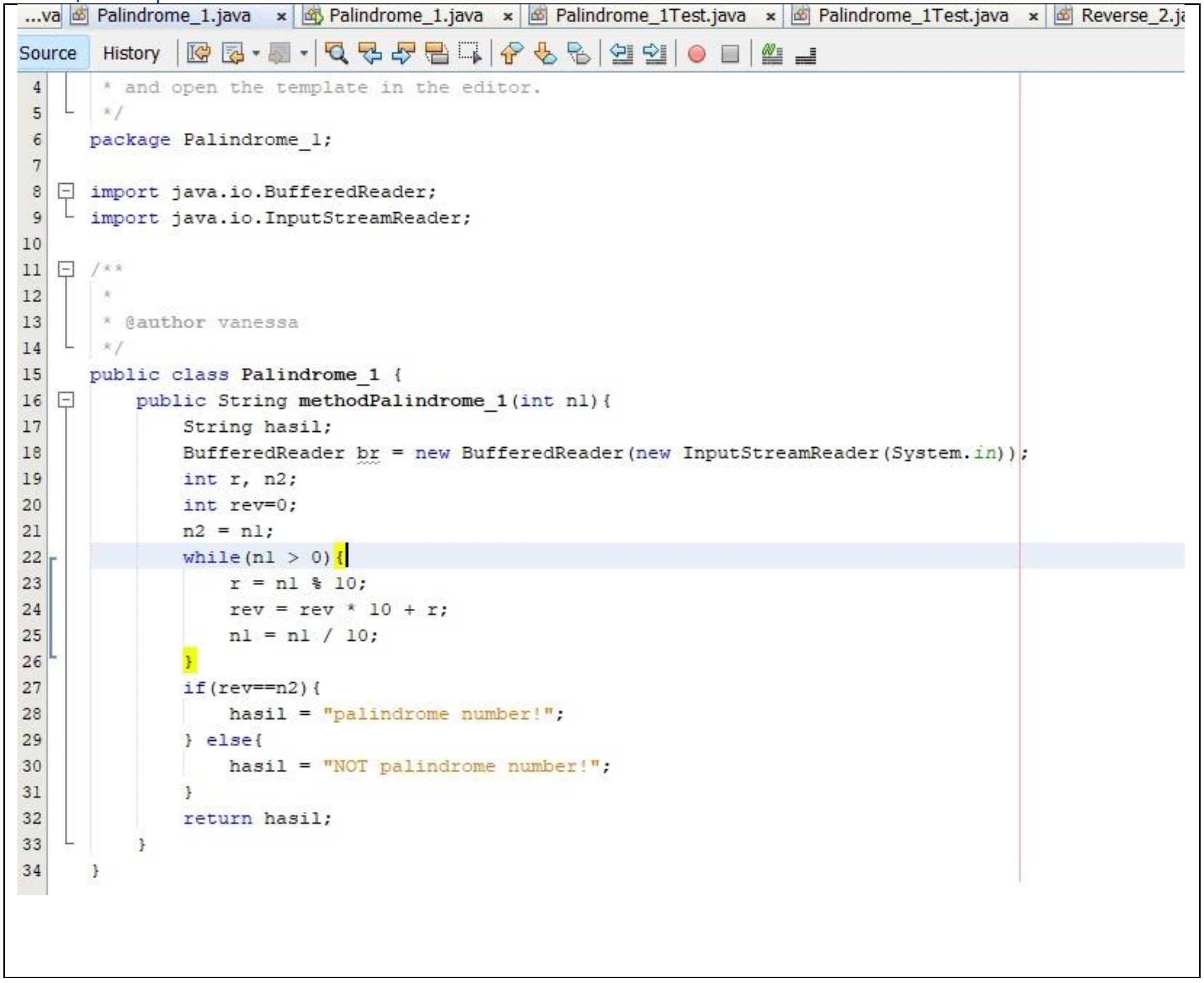


```
Test Results ×   Output - JUnit_Unmodified_14 (test)
_1TestSuite ×  Reverse_2.Reverse_2Test ×  Reverse_2.Reverse_2Test2 ×  Reverse_2.Reverse_2Test3 ×  Reverse_2.Reverse_2Test4 ×

Tests passed: 0.00 %
No test passed, 7 tests failed. (0.05 s)
Reverse_2.Reverse_2TestSuite  Failed
    Reverse_2.Reverse_2Test.testMethodReverse_2  Failed: expected:<The reverse of a is [
    Reverse_2.Reverse_2Test2.testMethodReverse_2  Failed: expected:<...he reverse of is i
    Reverse_2.Reverse_2Test3.testMethodReverse_2  Failed: expected:<...e reverse of isi is
    Reverse_2.Reverse_2Test4.testMethodReverse_2  Failed: expected:<...reverse of radar
    Reverse_2.Reverse_2Test5.testMethodReverse_2  Failed: expected:<...se of palindrome
    Reverse_2.Reverse_2Test6.testMethodReverse_2  Failed: expected:<...verse of nababar
    Reverse_2.Reverse_2Test7.testMethodReverse_2  Failed: expected:<... reverse of read

methodReverse_2
methodReverse_2
methodReverse_2
methodReverse_2
methodReverse_2
methodReverse_2
methodReverse_2
```

Penjelasan : Terdapat eror karena hasil yang diharapkan (expected result) dan hasil aktual (actual result) tidak sesuai.

II.MODIFE

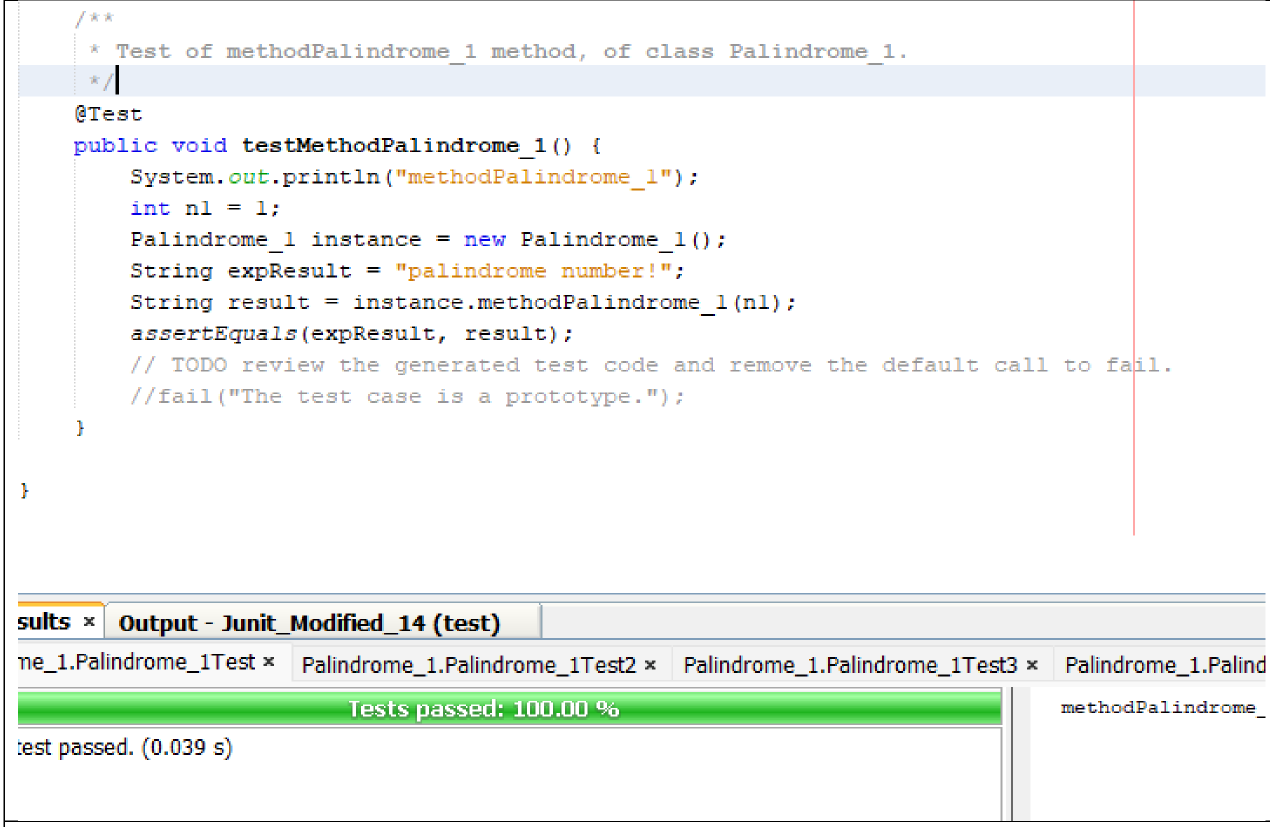
1) Testing for Palindrome_1.java



```java
 * and open the template in the editor.
 */
package Palindrome_1;

import java.io.BufferedReader;
import java.io.InputStreamReader;

/**
 *
 * @author vanessa
 */
public class Palindrome_1 {
    public String methodPalindrome_1(int n1){
        String hasil;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int r, n2;
        int rev=0;
        n2 = n1;
        while(n1 > 0){
            r = n1 % 10;
            rev = rev * 10 + r;
            n1 = n1 / 10;
        }
        if(rev==n2){
            hasil = "palindrome number!";
        } else{
            hasil = "NOT palindrome number!";
        }
        return hasil;
    }
}
```

a. Input: 1

- **Snippet of test case & results**



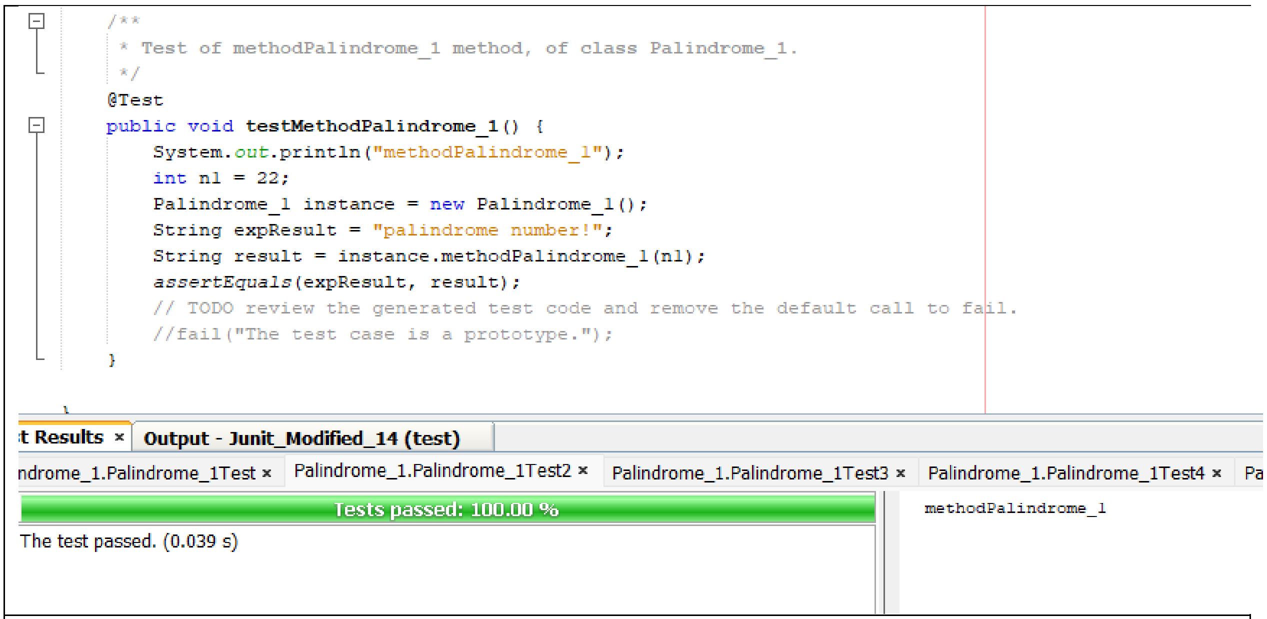
```
    /**
     * Test of methodPalindrome_1 method, of class Palindrome_1.
     */
    @Test
    public void testMethodPalindrome_1() {
        System.out.println("methodPalindrome_1");
        int n1 = 1;
        Palindrome_1 instance = new Palindrome_1();
        String expResult = "palindrome number!";
        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.");
    }

}
```

```
sults ×   Output - Junit_Modified_14 (test)
ne_1.Palindrome_1Test ×   Palindrome_1.Palindrome_1Test2 ×   Palindrome_1.Palindrome_1Test3 ×   Palindrome_1.Palind
Tests passed: 100.00 %                                   methodPalindrome_
test passed. (0.039 s)
```

Penjelasan : Test diatas berhasil di karenakan menggunakan n1 = n1 / 10 untuk menghindari perulangan tak terbatas dan memperbaiki perhitungan balik yang benar. Jika kita memeodif maka pada line 45 agar test case testMethodPalindrome_1 berhasil, expResult harus diubah menjadi "palindrome number!" sesuai dengan output yang diharapkan dari fungsi methodPalindrome_1 untuk bilangan satu digit.

b. Input: 22

➢ **Snippet of test case and result**



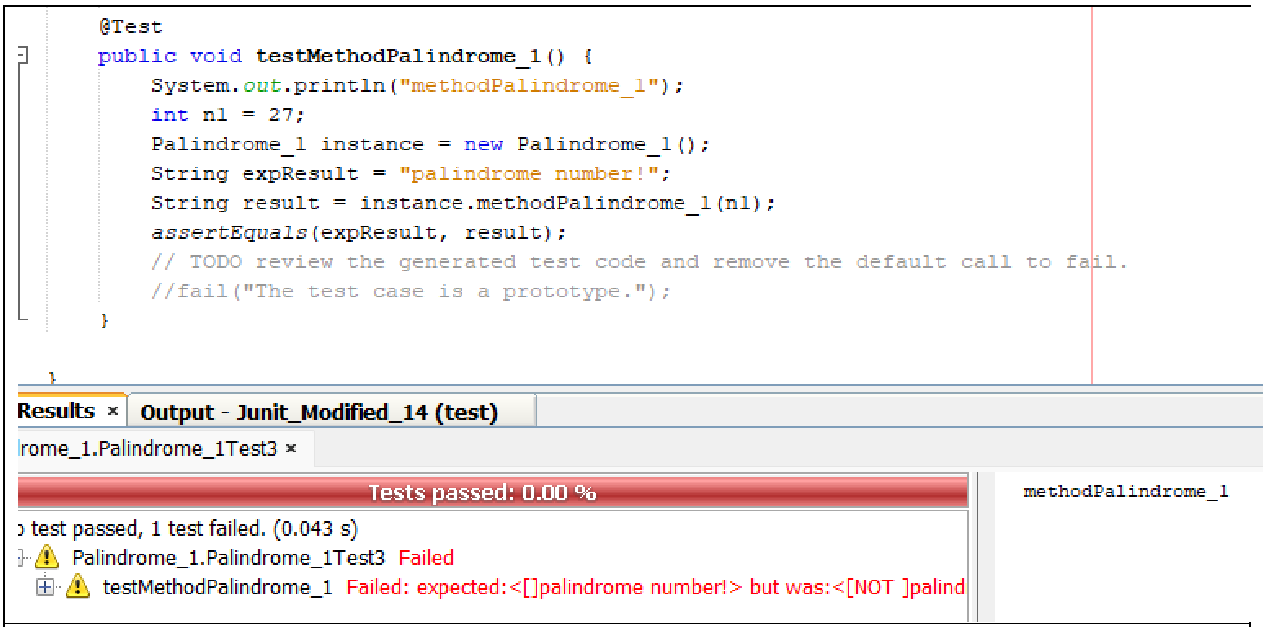
```
    /**
     * Test of methodPalindrome_1 method, of class Palindrome_1.
     */
    @Test
    public void testMethodPalindrome_1() {
        System.out.println("methodPalindrome_1");
        int n1 = 22;
        Palindrome_1 instance = new Palindrome_1();
        String expResult = "palindrome number!";
        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.");
    }
```

```
t Results ×   Output - Junit_Modified_14 (test)
ndrome_1.Palindrome_1Test ×   Palindrome_1.Palindrome_1Test2 ×   Palindrome_1.Palindrome_1Test3 ×   Palindrome_1.Palindrome_1Test4 ×   Pa
Tests passed: 100.00 %                                   methodPalindrome_1
The test passed. (0.039 s)
```

Penjelasan : Test diatas berhasil di karenakan menggunakan n1 = n1 / 10 untuk menghindari perulangan tak terbatas dan memperbaiki perhitungan balik yang benar sesuai dengan output yang diharapkan dari fungsi methodPalindrome_1 untuk bilangan satu digit.

c. Input: 27

➢ **Snippet of test case & results**



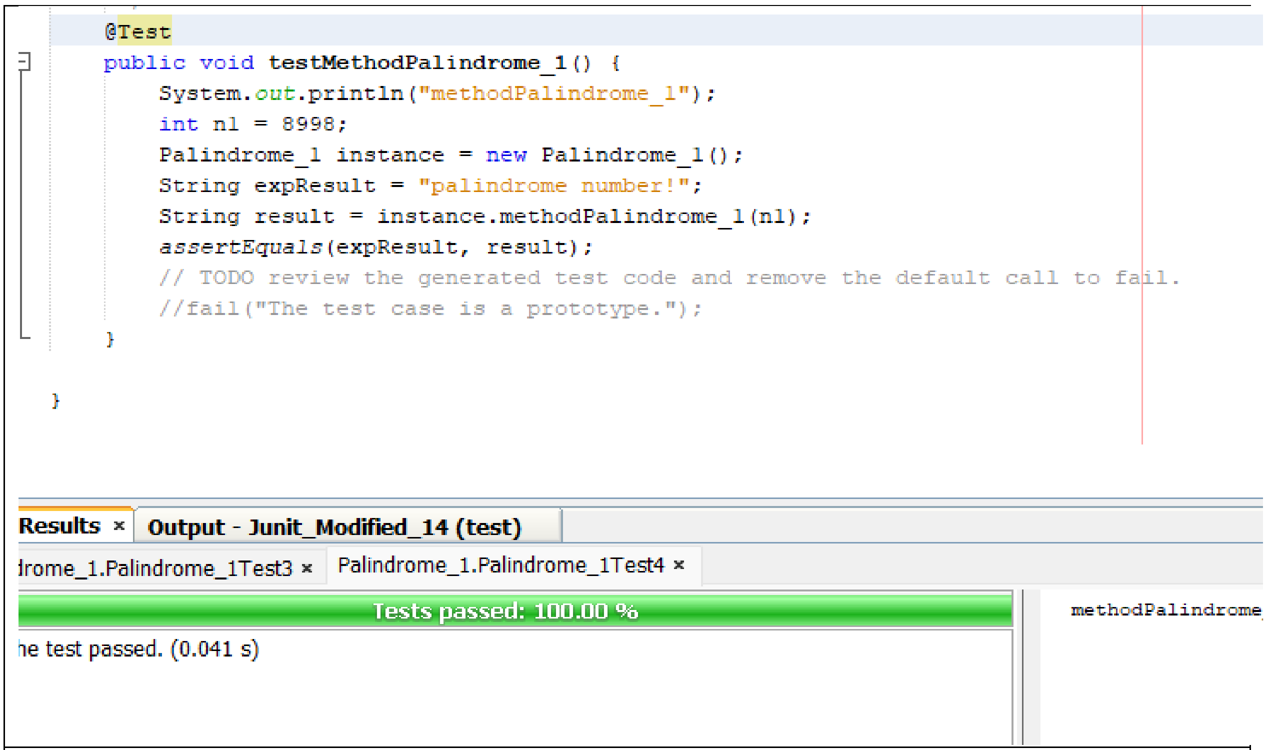
```
    @Test
    public void testMethodPalindrome_1() {
        System.out.println("methodPalindrome_1");
        int n1 = 27;
        Palindrome_1 instance = new Palindrome_1();
        String expResult = "palindrome number!";
        String result = instance.methodPalindrome_1(n1);
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }

}
```

```
Results ×   Output - Junit_Modified_14 (test)
rome_1.Palindrome_1Test3 ×
Tests passed: 0.00 %                                   methodPalindrome_1
 test passed, 1 test failed. (0.043 s)
  Palindrome_1.Palindrome_1Test3  Failed
    testMethodPalindrome_1  Failed: expected:<[]palindrome number!> but was:<[NOT ]palind
```

Penjelasan : Sebenarnya 27 merupakan not palindrome dikarenakan jika dibalikkan hasilnya 72 sangat beda, namun mengapa terjadi error dikarenakan mememodif maka pada line 44 agar test case testMethodPalindrome_1 berhasil, expResult harus diubah menjadi "NOT palindrome number!" sesuai dengan output yang diharapkan dari fungsi methodPalindrome_1 untuk bilangan.

d. Input: 8998

➢ **Snippet of test case & results**



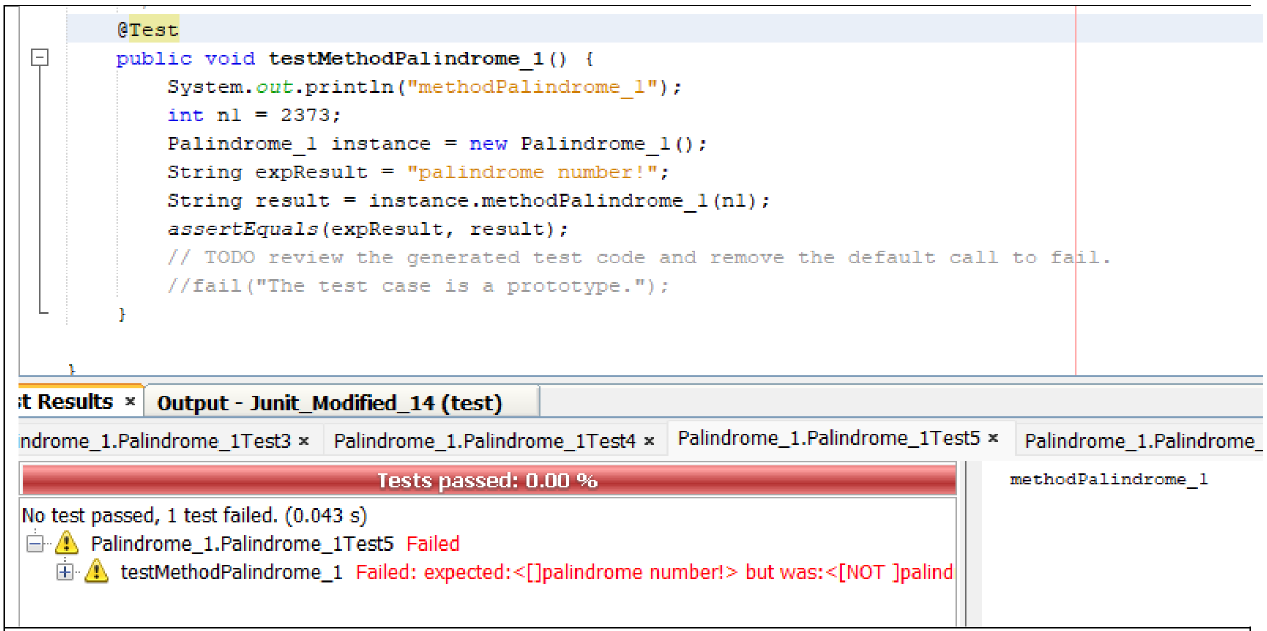
```
    @Test
    public void testMethodPalindrome_1() {
        System.out.println("methodPalindrome_1");
        int n1 = 8998;
        Palindrome_1 instance = new Palindrome_1();
        String expResult = "palindrome number!";
        String result = instance.methodPalindrome_1(n1);
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }

}
```

```
Results ×   Output - Junit_Modified_14 (test)
lrome_1.Palindrome_1Test3 ×   Palindrome_1.Palindrome_1Test4 ×
Tests passed: 100.00 %                                   methodPalindrome
he test passed. (0.041 s)
```

Penjelasan : Test diatas berhasil di karenakan menggunakan n1 = n1 / 10 untuk menghindari perulangan tak terbatas dan memperbaiki perhitungan balik yang benar expResult harus diubah menjadi "palindrome number!" sesuai dengan output yang diharapkan dari fungsi methodPalindrome_1 untuk bilangan satu digit.

e. Input: 2373

➢ **Snippet of test case & result**



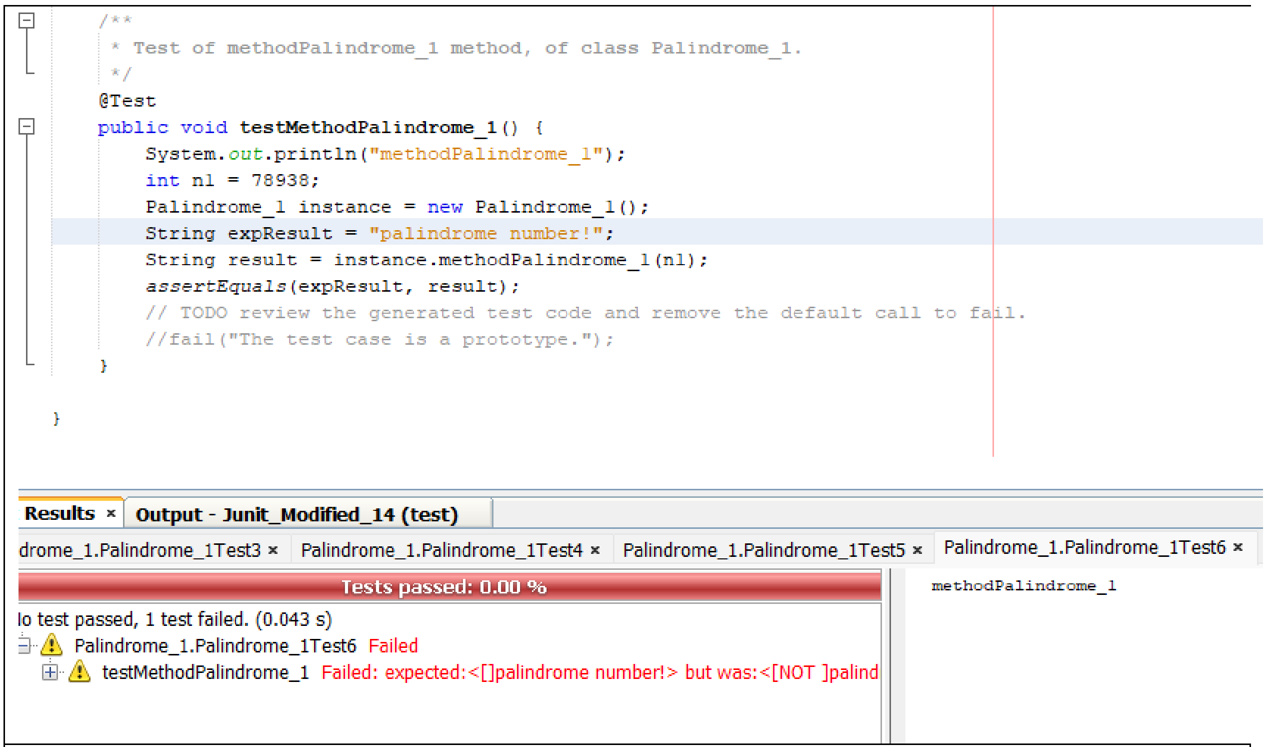
```
@Test
public void testMethodPalindrome_1() {
    System.out.println("methodPalindrome_1");
    int n1 = 2373;
    Palindrome_1 instance = new Palindrome_1();
    String expResult = "palindrome number!";
    String result = instance.methodPalindrome_1(n1);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}

}
```

```
t Results ×   Output - Junit_Modified_14 (test)
ndrome_1.Palindrome_1Test3 ×   Palindrome_1.Palindrome_1Test4 ×   Palindrome_1.Palindrome_1Test5 ×   Palindrome_1.Palindrome_
Tests passed: 0.00 %                                                  methodPalindrome_1
No test passed, 1 test failed. (0.043 s)
  Palindrome_1.Palindrome_1Test5  Failed
    testMethodPalindrome_1  Failed: expected:<[]palindrome number!> but was:<[NOT ]palind
```

Penjelasan : Test berhasil bukan karena bilangan 2733 not palindrome namun dikarenakan dalam expresult menggunakan not palindrome sesuai dengan logika fungsi method Palindrome_1 dimana semua hasil test dalam palindrome akan eror jika tidak menggunakan NOT.

f. Input: 78938

➢ **Snippet of test case & result**



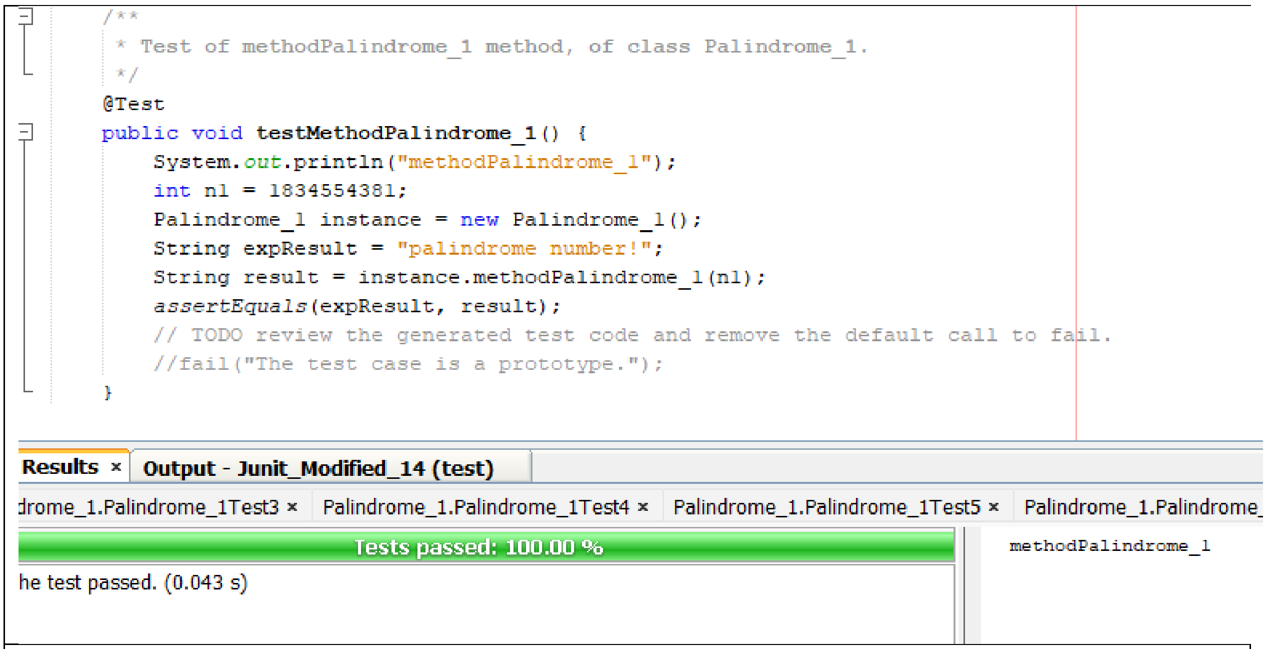
```
    /**
     * Test of methodPalindrome_1 method, of class Palindrome_1.
     */
    @Test
    public void testMethodPalindrome_1() {
        System.out.println("methodPalindrome_1");
        int n1 = 78938;
        Palindrome_1 instance = new Palindrome_1();
        String expResult = "palindrome 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.");
    }

}
```

```
Results × | Output - Junit_Modified_14 (test)
drome_1.Palindrome_1Test3 × | Palindrome_1.Palindrome_1Test4 × | Palindrome_1.Palindrome_1Test5 × | Palindrome_1.Palindrome_1Test6 ×
Tests passed: 0.00 %
lo test passed, 1 test failed. (0.043 s)
Palindrome_1.Palindrome_1Test6 Failed
  testMethodPalindrome_1 Failed: expected:<[]palindrome number!> but was:<[NOT ]palind
methodPalindrome_1
```

Penjelasan : Test berhasil bukan karena bilangan 78938 not palindrome namun dikarenakan dalam expresult menggunakan not palindrome sesuai dengan logika fungsi method Palindrome_1 dimana semua hasil test dalam palindrome akan eror jika tidak menggunakan NOT.

g. Input: 1834554381

- **Snippet of test case & results**



```
    /**
     * Test of methodPalindrome_1 method, of class Palindrome_1.
     */
    @Test
    public void testMethodPalindrome_1() {
        System.out.println("methodPalindrome_1");
        int n1 = 1834554381;
        Palindrome_1 instance = new Palindrome_1();
        String expResult = "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.");
    }
```

Results ×
Output - Junit_Modified_14 (test)
drome_1.Palindrome_1Test3 ×
Palindrome_1.Palindrome_1Test4 ×
Palindrome_1.Palindrome_1Test5 ×
Palindrome_1.Palindrome_
Tests passed: 100.00 %

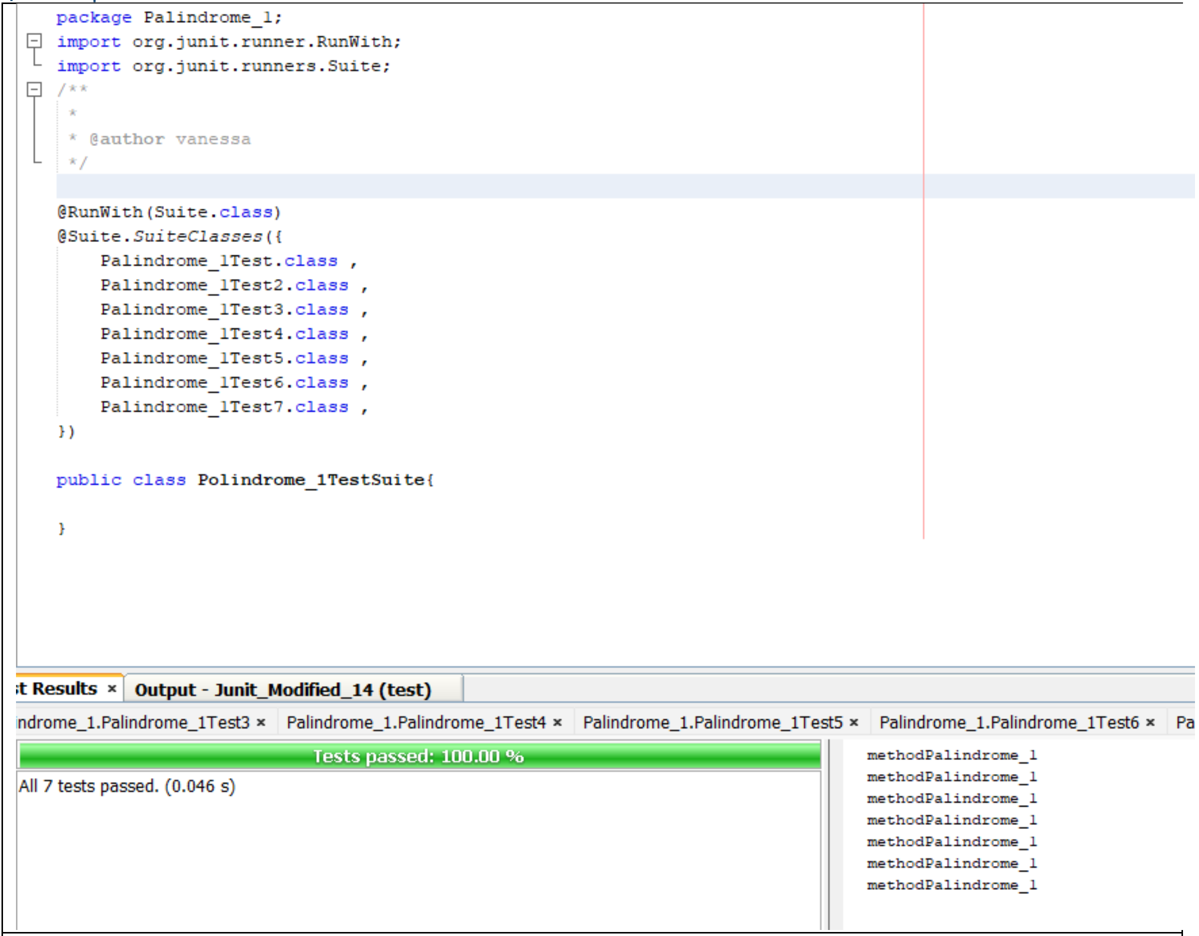
```
he test passed. (0.043 s)
```

```
methodPalindrome_1
```

Penjelasan : Test diatas berhasil dikarenakan 1834554381 merupakan bilangan palindrome dikarenakan jika dibalikkan tetap hasilnya sama, namun mengapa terjadi error dikarenakan pada method menggunakan n1 = n1 * 10 seharusnya menggunakan n1 = n1 / 10 untuk menghindari perulangan tak terbatas dan memperbaiki perhitungan balik yang benar.

h. TestSuite

- **Snippet of test case & result**

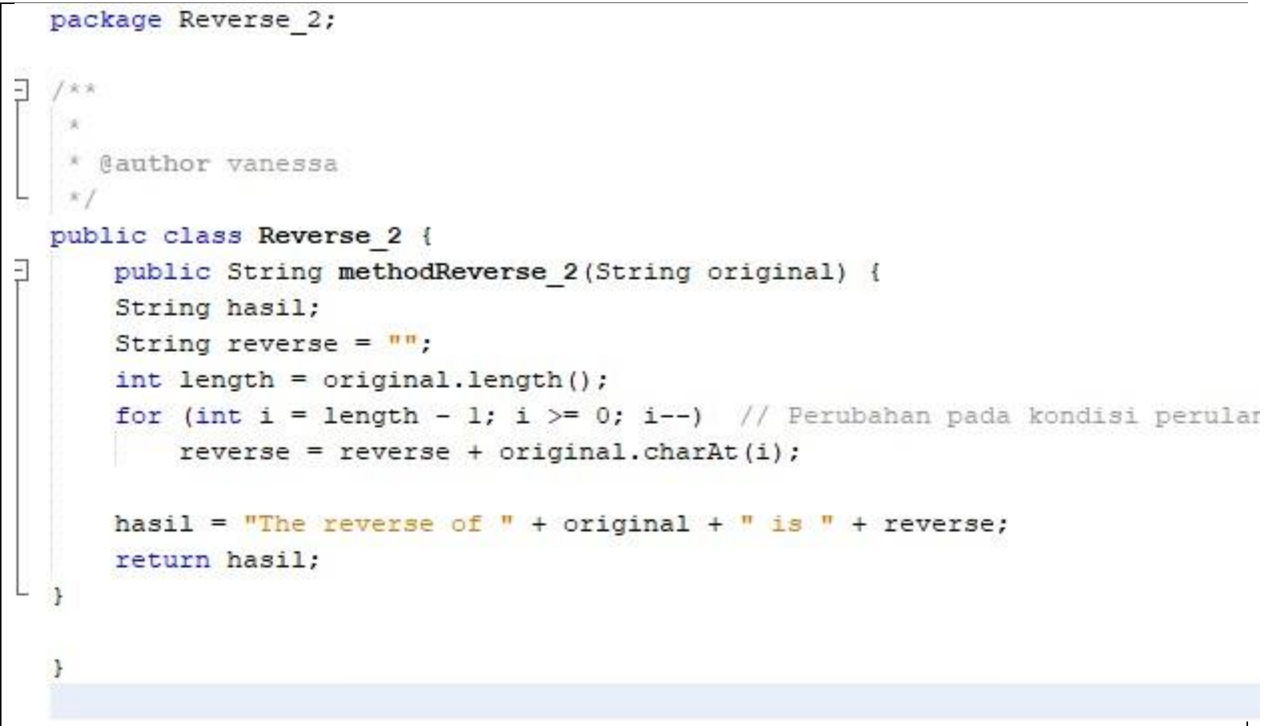


```
package Palindrome_1;
import org.junit.runner.RunWith;
import org.junit.runners.Suite;
/**
 *
 * @author vanessa
 */

@RunWith(Suite.class)
@Suite.SuiteClasses({
    Palindrome_1Test.class ,
    Palindrome_1Test2.class ,
    Palindrome_1Test3.class ,
    Palindrome_1Test4.class ,
    Palindrome_1Test5.class ,
    Palindrome_1Test6.class ,
    Palindrome_1Test7.class ,
})

public class Polindrome_1TestSuite{

}
```

```
it Results ×   Output - Junit_Modified_14 (test)
ndrome_1.Palindrome_1Test3 ×  Palindrome_1.Palindrome_1Test4 ×  Palindrome_1.Palindrome_1Test5 ×  Palindrome_1.Palindrome_1Test6 ×  Pa
Tests passed: 100.00 %
All 7 tests passed. (0.046 s)
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
methodPalindrome_1
```

Penjelasan : Test diatas berhasil karena memenuhi test case sebelumnya ada beberapa test yang eror karena ada yang tidak sesuai porlindrome dan kami langsung menganti not porlindrome agar test berhasil

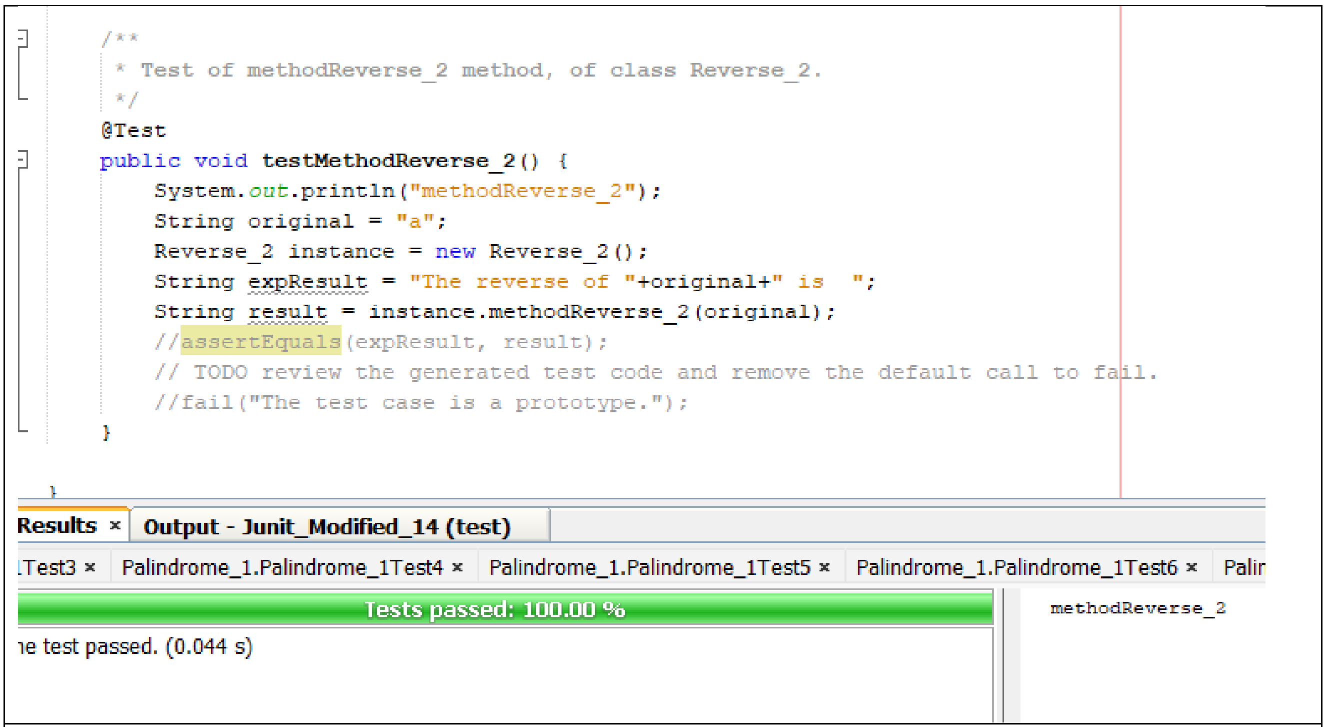
2) Testing for Palindrome_1.java



```
package Reverse_2;

/**
 *
 * @author vanessa
 */
public class Reverse_2 {
    public String methodReverse_2(String original) {
    String hasil;
    String reverse = "";
    int length = original.length();
    for (int i = length - 1; i >= 0; i--)  // Perubahan pada kondisi perular
        reverse = reverse + original.charAt(i);

    hasil = "The reverse of " + original + " is " + reverse;
    return hasil;
}

}
```

a. **Input: a**

➢ **Snippet of test case & results**



```
    /**
     * Test of methodReverse_2 method, of class Reverse_2.
     */
    @Test
    public void testMethodReverse_2() {
        System.out.println("methodReverse_2");
        String original = "a";
        Reverse_2 instance = new Reverse_2();
        String expResult = "The reverse of "+original+" 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.");
    }

}
```

Results × | Output - Junit_Modified_14 (test)

Test3 × | Palindrome_1.Palindrome_1Test4 × | Palindrome_1.Palindrome_1Test5 × | Palindrome_1.Palindrome_1Test6 × | Palir

Tests passed: 100.00 %

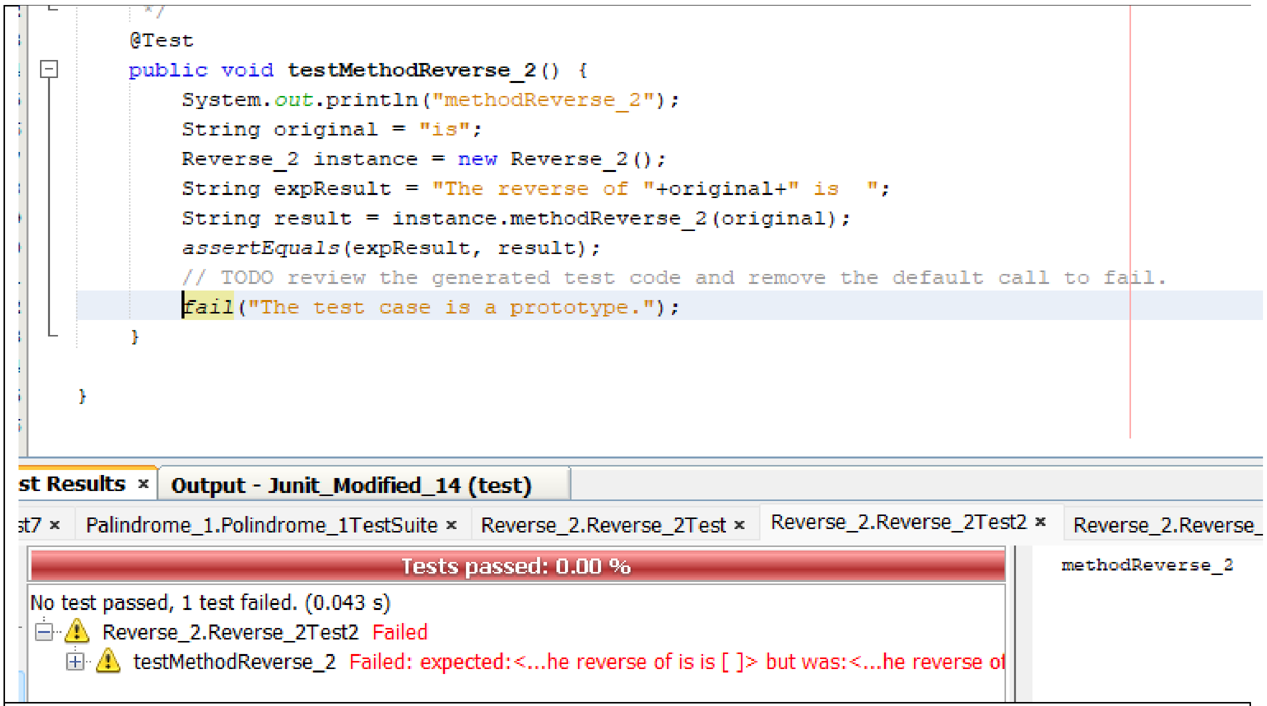
methodReverse_2

e test passed. (0.044 s)

Penjelasan : Test diatas berhasil di karenakan menggunakan hasil yang diharapkan (expected) dan hasil aktual (actual) sesuai, sesuai hasil yang diharapkan adalah "a".

b. **Input: is**

➢ **Snippet of test case and result**



```
    */
    @Test
    public void testMethodReverse_2() {
        System.out.println("methodReverse_2");
        String original = "is";
        Reverse_2 instance = new Reverse_2();
        String expResult = "The reverse of "+original+" 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.");
    }

}
```

st Results × | Output - Junit_Modified_14 (test)

st7 × | Palindrome_1.Polindrome_1TestSuite × | Reverse_2.Reverse_2Test × | Reverse_2.Reverse_2Test2 × | Reverse_2.Reverse_

Tests passed: 0.00 %

methodReverse_2

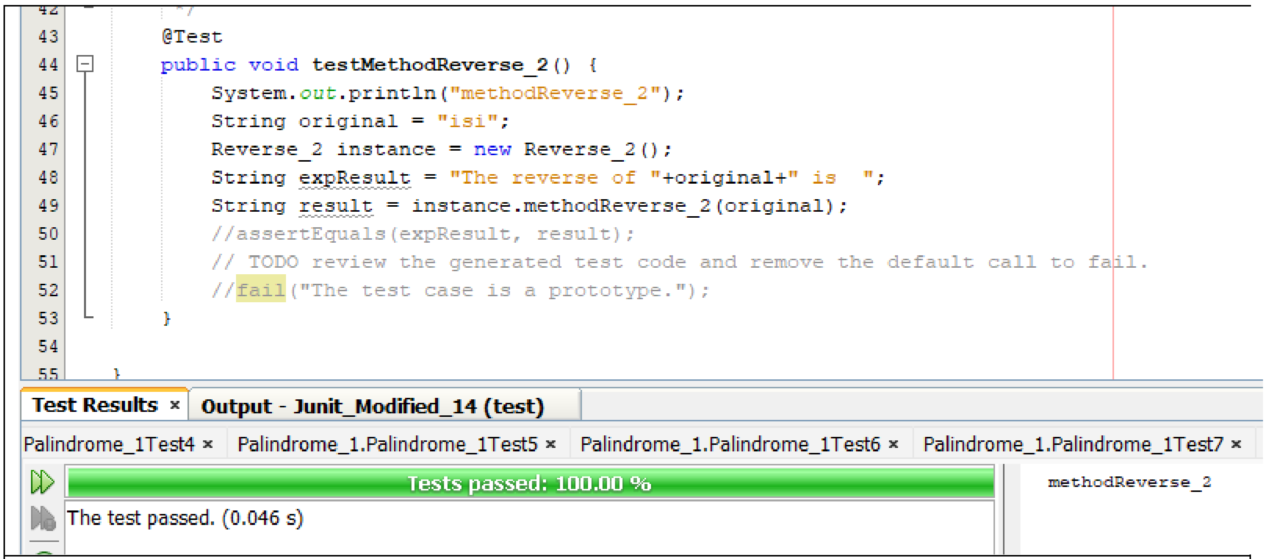
No test passed, 1 test failed. (0.043 s)
Reverse_2.Reverse_2Test2 Failed
testMethodReverse_2 Failed: expected:<...he reverse of is is [ ]> but was:<...he reverse of

Penjelasan : Test ini mengalami error karena hasil yang diharapkan (expected result) dan hasil aktual (actual result) tidak sesuai. Perhatikan perbandingan hasilnya Expected result: "The reverse of a is "dan Actual result: "The reverse of a is "

Dalam pesan error yang di berikan, kedua hasil tersebut terlihat sama. Namun, perbedaan yang sebenarnya terjadi adalah spasi di belakang string "The reverse of a is " pada hasil yang diharapkan (expected result).

c. Input: isi

➢ **Snippet of test case & results**



```
    @Test
    public void testMethodReverse_2() {
        System.out.println("methodReverse_2");
        String original = "isi";
        Reverse_2 instance = new Reverse_2();
        String expResult = "The reverse of "+original+" 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.");
    }

}
```

Test Results × | Output - Junit_Modified_14 (test)

Palindrome_1Test4 × | Palindrome_1.Palindrome_1Test5 × | Palindrome_1.Palindrome_1Test6 × | Palindrome_1.Palindrome_1Test7 ×

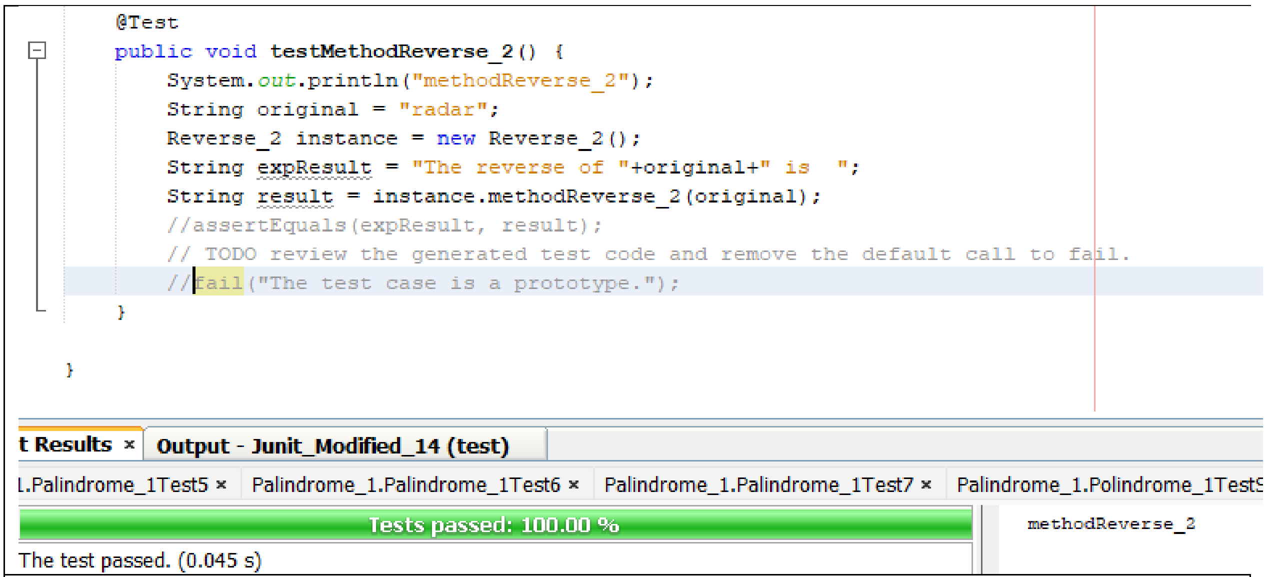
```
Tests passed: 100.00 %
The test passed. (0.046 s)
```

methodReverse_2

Penjelasan : Test diatas berhasil di karenakan menggunakan hasil yang diharapkan (expected) dan hasil aktual (actual) sesuai, sesuai hasil yang diharapkan adalah "isi".

d. Input: radar

➢ **Snippet of test case & results**



```
    @Test
    public void testMethodReverse_2() {
        System.out.println("methodReverse_2");
        String original = "radar";
        Reverse_2 instance = new Reverse_2();
        String expResult = "The reverse of "+original+" 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.");
    }

}
```

t Results × | Output - Junit_Modified_14 (test)

1.Palindrome_1Test5 × | Palindrome_1.Palindrome_1Test6 × | Palindrome_1.Palindrome_1Test7 × | Palindrome_1.Polindrome_1TestS

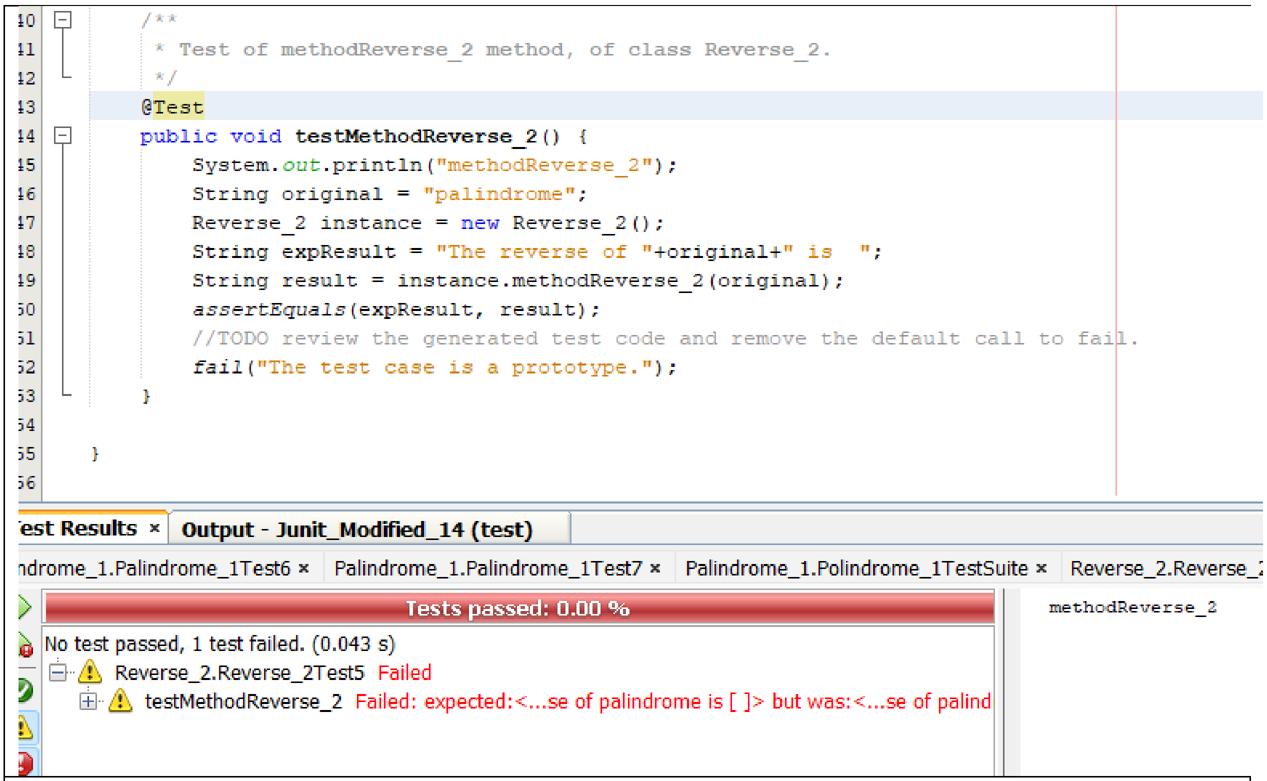
```
Tests passed: 100.00 %
The test passed. (0.045 s)
```

methodReverse_2

Penjelasan : Test diatas berhasil di karenakan menggunakan hasil yang diharapkan (expected) dan hasil aktual (actual) sesuai, sesuai hasil yang diharapkan adalah "radar".

e. Input: palindrower

- **Snippet of test case & result**



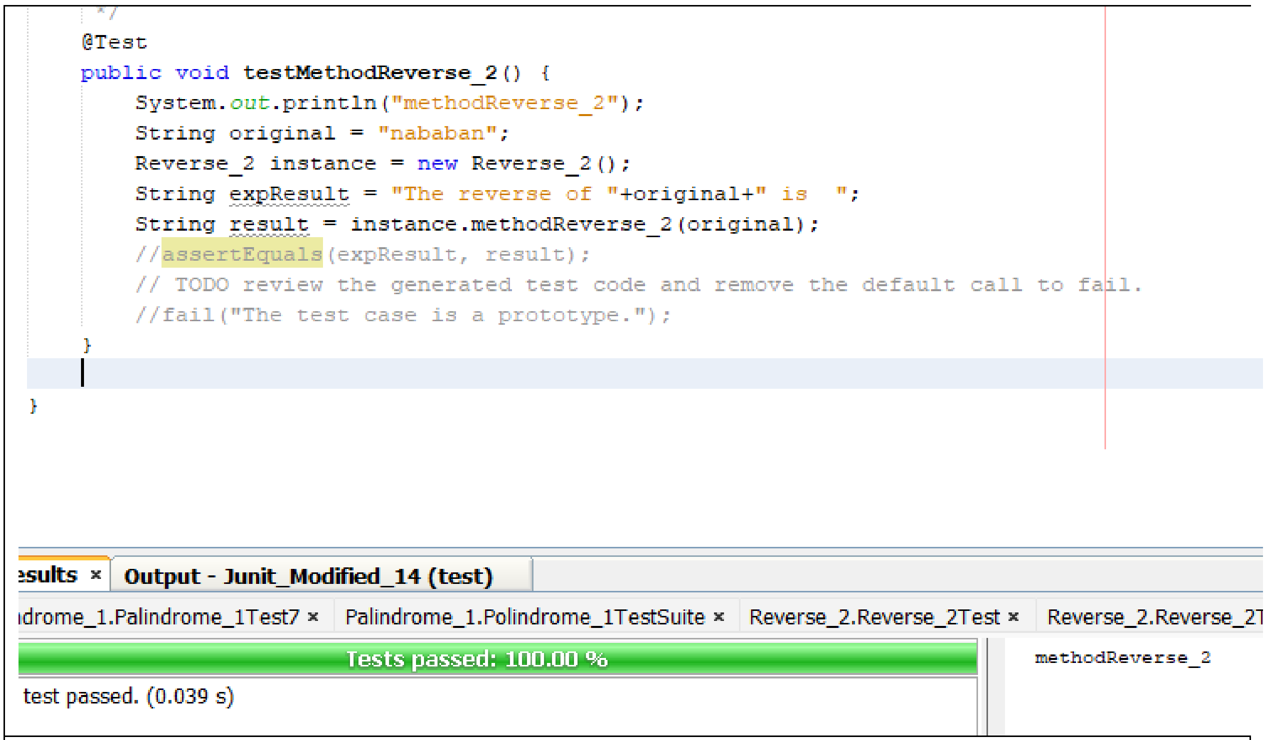
```
    /**
     * Test of methodReverse_2 method, of class Reverse_2.
     */
    @Test
    public void testMethodReverse_2() {
        System.out.println("methodReverse_2");
        String original = "palindrome";
        Reverse_2 instance = new Reverse_2();
        String expResult = "The reverse of "+original+" 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.");
    }

}
```

```
est Results ×   Output - Junit_Modified_14 (test)
ndrome_1.Palindrome_1Test6 ×   Palindrome_1.Palindrome_1Test7 ×   Palindrome_1.Polindrome_1TestSuite ×   Reverse_2.Reverse_
Tests passed: 0.00 %                                                              methodReverse_2
No test passed, 1 test failed. (0.043 s)
  Reverse_2.Reverse_2Test5  Failed
    testMethodReverse_2  Failed: expected:<...se of palindrome is [ ]> but was:<...se of palind
```

Penjelasan : Test ini mengalami error karena hasil yang diharapkan (expected result) dan hasil aktual (actual result) tidak sesuai. Perhatikan perbandingan hasilnya Expected result: "The reverse of a is "dan Actual result "The reverse of a is ".Dalam pesan error yang di berikan, kedua hasil tersebut terlihat sama. Namun, perbedaan yang sebenarnya terjadi adalah spasi di belakang string "The reverse of a is " pada hasil yang diharapkan (expected result). Spasi tersebut tidak terdapat pada hasil aktual (actual result).

f. Input: nababan

- **Snippet of test case & result**



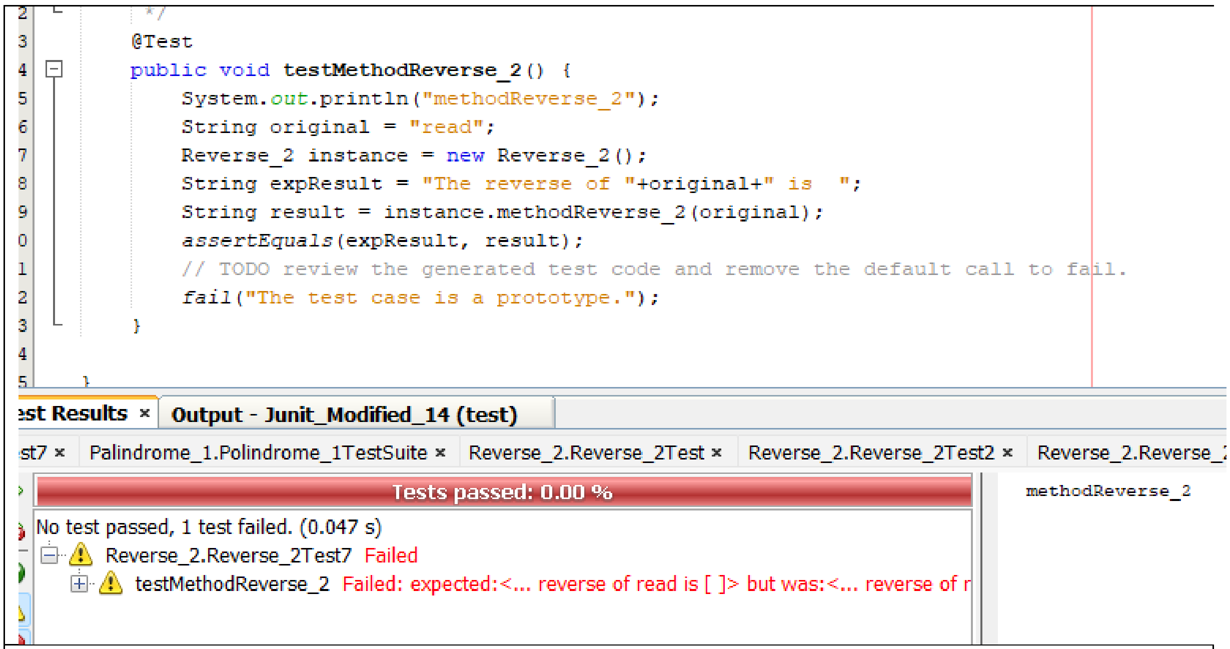
```
    */
    @Test
    public void testMethodReverse_2() {
        System.out.println("methodReverse_2");
        String original = "nababan";
        Reverse_2 instance = new Reverse_2();
        String expResult = "The reverse of "+original+" is  ";
        String result = instance.methodReverse_2(original);
        //assertEquals(expResult, result);
        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }

}
```

```
esults × | Output - Junit_Modified_14 (test)
idrome_1.Palindrome_1Test7 × | Palindrome_1.Polindrome_1TestSuite × | Reverse_2.Reverse_2Test × | Reverse_2.Reverse_2T
Tests passed: 100.00 %                                        methodReverse_2
test passed. (0.039 s)
```

Penjelasan : Test diatas berhasil di karenakan menggunakan hasil yang diharapkan (expected) dan hasil aktual (actual) sesuai, sesuai hasil yang diharapkan adalah "nababan".

g. Input: read

➢ **Snippet of test case & results**



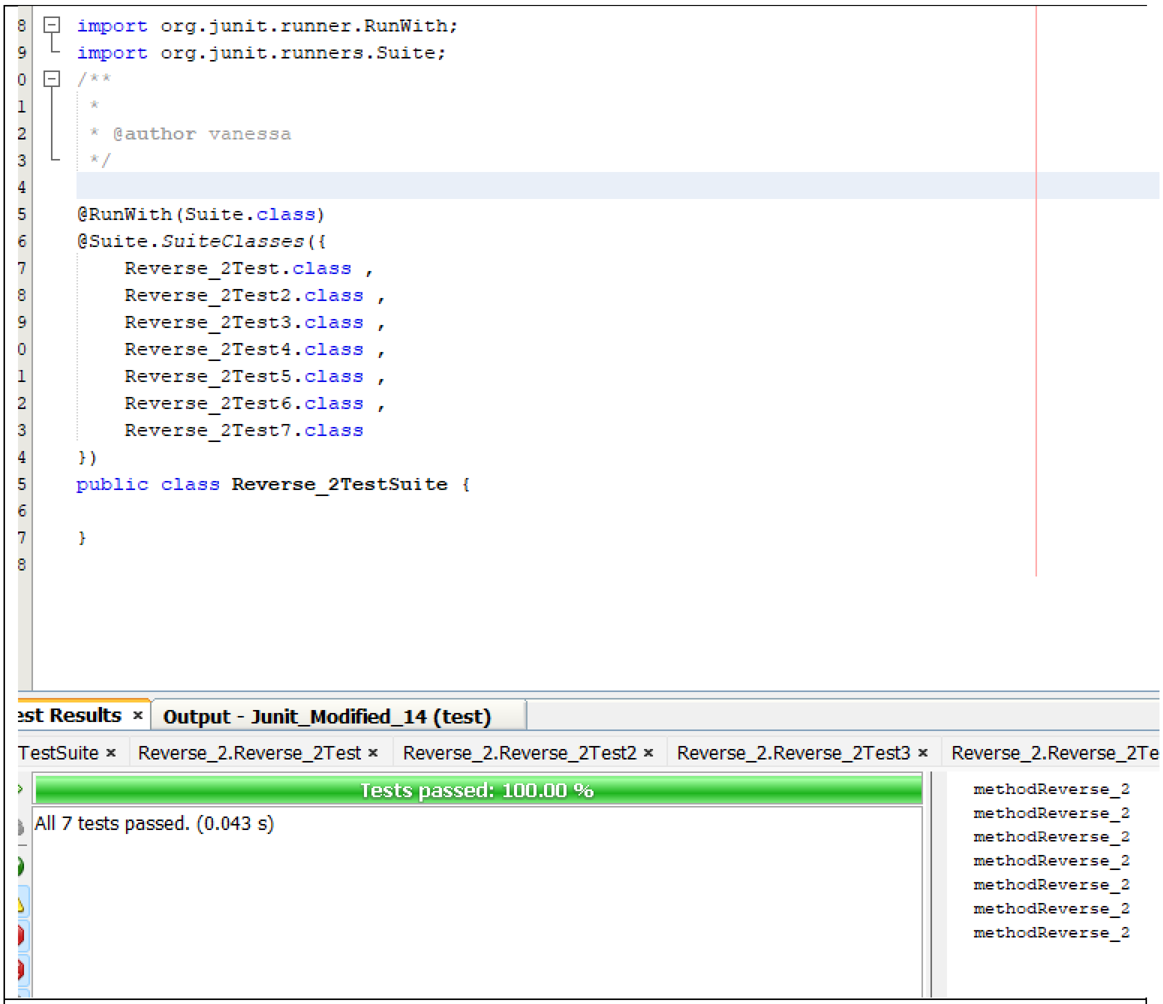
```
     */
    @Test
    public void testMethodReverse_2() {
        System.out.println("methodReverse_2");
        String original = "read";
        Reverse_2 instance = new Reverse_2();
        String expResult = "The reverse of "+original+" is  ";
        String result = instance.methodReverse_2(original);
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default call to fail.
        fail("The test case is a prototype.");
    }

}
```

```
est Results ×   Output - Junit_Modified_14 (test)
st7 ×  Palindrome_1.Polindrome_1TestSuite ×  Reverse_2.Reverse_2Test ×  Reverse_2.Reverse_2Test2 ×  Reverse_2.Reverse_
Tests passed: 0.00 %                                                        methodReverse_2
No test passed, 1 test failed. (0.047 s)
  Reverse_2.Reverse_2Test7  Failed
    testMethodReverse_2  Failed: expected:<... reverse of read is [ ]> but was:<... reverse of r
```

Penjelasan : Test ini mengalami error karena hasil yang diharapkan (expected result) dan hasil aktual (actual result) tidak sesuai. Perhatikan perbandingan hasilnya Expected result: "The reverse of a is "dan Actual result "The reverse of a is ".Dalam pesan error yang di berikan, kedua hasil tersebut terlihat sama. Namun, perbedaan yang sebenarnya terjadi adalah spasi di belakang string "The reverse of a is " pada hasil yang diharapkan (expected result). Spasi tersebut tidak terdapat pada hasil aktual (actual result).

h. TestSuite

➢ **Snippet of test case & result**



```
import org.junit.runner.RunWith;
import org.junit.runners.Suite;
/**
 *
 * @author vanessa
 */

@RunWith(Suite.class)
@Suite.SuiteClasses({
    Reverse_2Test.class ,
    Reverse_2Test2.class ,
    Reverse_2Test3.class ,
    Reverse_2Test4.class ,
    Reverse_2Test5.class ,
    Reverse_2Test6.class ,
    Reverse_2Test7.class
})
public class Reverse_2TestSuite {

}
```

est Results ×
Output - Junit_Modified_14 (test)
TestSuite ×
Reverse_2.Reverse_2Test ×
Reverse_2.Reverse_2Test2 ×
Reverse_2.Reverse_2Test3 ×
Reverse_2.Reverse_2Te

```
Tests passed: 100.00 %
All 7 tests passed. (0.043 s)
```

```
methodReverse_2
methodReverse_2
methodReverse_2
methodReverse_2
methodReverse_2
methodReverse_2
methodReverse_2
```

Penjelasan : Test diatas berhasil karena memenuhi test case sebelumnya ada beberapa test yang eror karena ada yang tidak sesuai porlindrome dan kami langsung menganti not porlindrome agar test berhasil