

Name: Andi Dwi Saputro

Date: April 24th, 2020

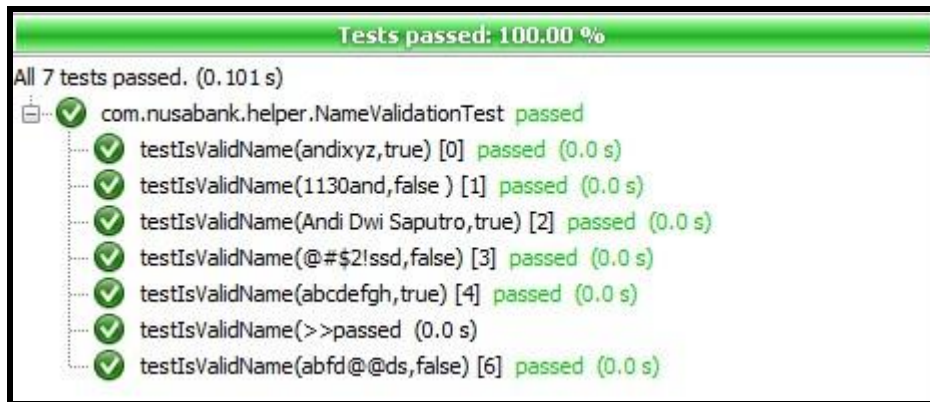
1. NameValidationTest Case

File: NameValidationTest.java

```
6 package com.nusabank.helper;
7
8 import org.junit.Test;
9 import static org.junit.Assert.*;
10 import junitparams.*;
11
12 import org.junit.runner.RunWith;
13
14 import com.nusabank.helper.Validation;
15
16 /**
17  *
18  * @author User
19  */
20 @RunWith(JUnitParamsRunner.class)
21 public class NameValidationTest {
22     Validation validTest = new Validation();
23
24     @Test
25     @FileParameters("resources/data_name.csv")
26     public void testIsValidName(final String a, final boolean b) {
27         System.out.println("isValidName");
28         String val = a;
29         boolean expectedResult = b;
30         boolean result = validTest.isValidName(val);
31         assertEquals(expectedResult, result);
32     }
33 }
```

Saya menggunakan library tambahan **JUnitParams** untuk melakukan parsing file CSV menjadi parameter untuk proses asersi dan testing. Disini result nya berupa Boolean apabila result isValidName(val) sama dengan expected result yang ada di CSV maka test case results nya akan passed semua.

Test Case Results:



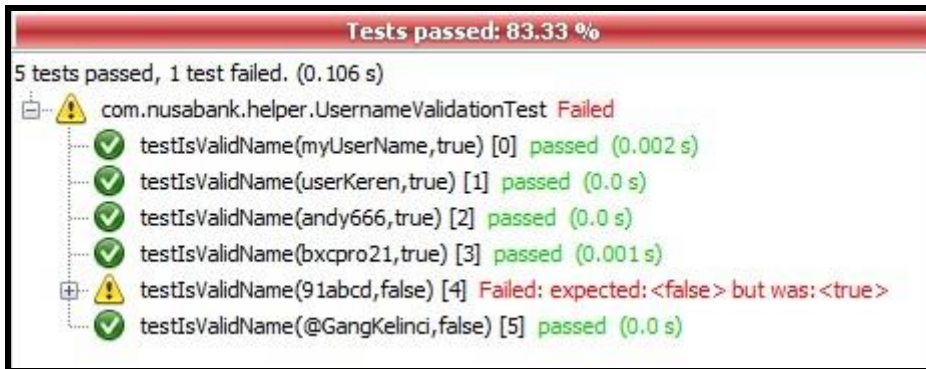
Semua parameter menghasilkan test passed karena result Boolean dari method `isValidName` bernilai sama dengan variable `expResult` sebagai nilai yang diharapkan untuk sama nilainya.

2. UsernameValidationTest Case

File: `UsernameValidationTest.java`

```
6 package com.nusabank.helper;
7
8 import junitparams.FileParameters;
9 import junitparams.JUnit4ParamsRunner;
10 import static org.junit.Assert.assertEquals;
11 import org.junit.Test;
12 import org.junit.runner.RunWith;
13
14 /**
15  *
16  * @author User
17  */
18 @RunWith(JUnit4ParamsRunner.class)
19 public class UsernameValidationTest {
20     Validation validTest = new Validation();
21
22     @Test
23     @FileParameters("resources/data_username.csv")
24     public void testIsValidUsername(final String a, final boolean b){
25         System.out.println("isValidUsername");
26         String val = a;
27         boolean expResult = b;
28         boolean result = validTest.isValidUsername(val);
29         assertEquals(expResult, result);
30     }
31 }
```

Test Case Results:



Disini saya sengaja menambahkan kesalahan pada salah satu record data di CSV karena regex pattern `isValidUsername()` menanggapi true nilai 91abcd.

3. EmailValidationTest Case

File: `EmailValidationTest.java`

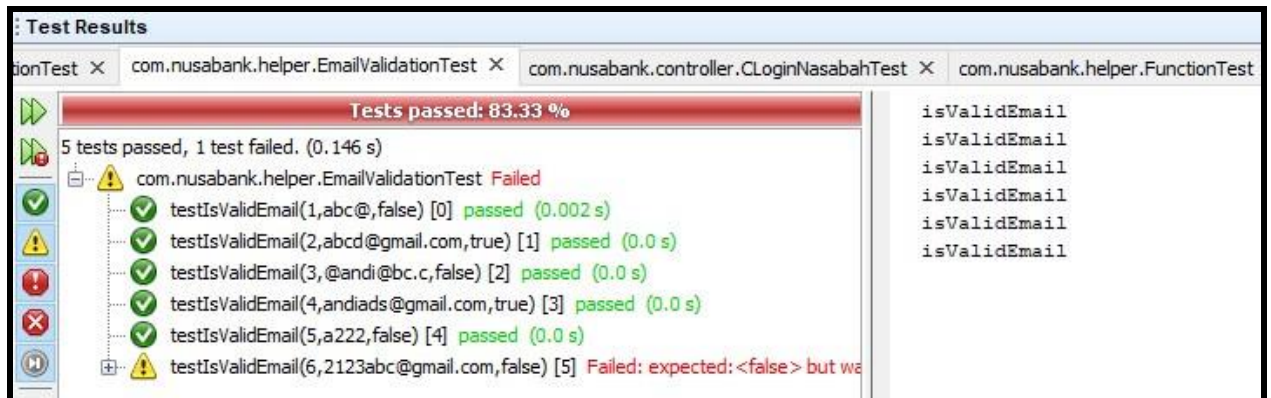
```
@RunWith(JUnitParamsRunner.class)
public class EmailValidationTest {

    Validation validTest = new Validation();

    public EmailValidationTest() {
    }

    @Test
    @FileParameters("resources/data_email.csv")
    public void testIsValidEmail(final int a, final String b, final boolean c) {
        System.out.println("isValidEmail");
        String val = b;
        boolean expResult = c;
        boolean result = validTest.isValidEmail(val);
        assertEquals(expResult, result);
    }
}
```

Test Case Results:



4. PasswordValidationTest Case

File: PasswordValidationTest.java

```

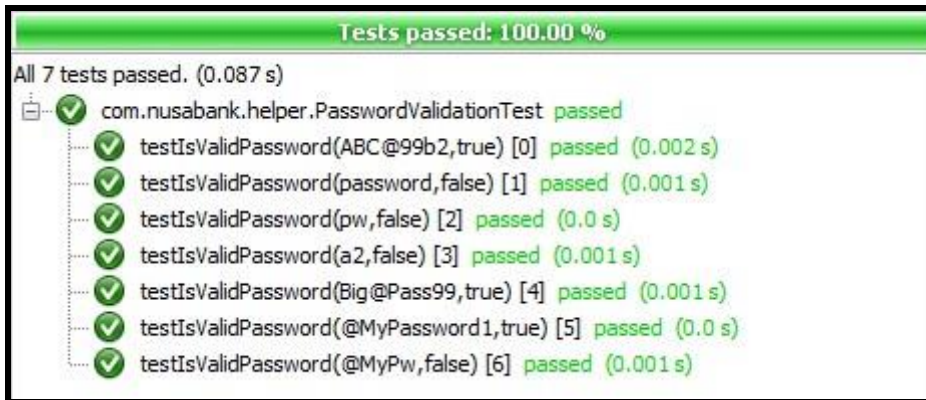
23  @RunWith(JUnitParamsRunner.class)
24  public class PasswordValidationTest {
25      Validation validTest = Validation.buildPasswordRule(true, true, true, 6, 12);
26
27      @Test
28      @FileParameters("resources/data_password.csv")
29      public void testIsValidPassword(final String a, final boolean b) {
30          System.out.println("isValidPassword");
31          String val = a;
32          boolean expectedResult = b;
33          boolean result = validTest.isValidPassword(val);
34          assertEquals(expectedResult, result);
35      }
36  }

```

Khusus untuk validasi password, harus melakukan setup password rule dengan `buildPasswordRule` dengan kriteria password:

- Harus memiliki minimal 1 angka
- Harus memiliki minimal 1 alphanumeric symbol
- Harus memiliki minimal 1 huruf besar
- Harus memiliki panjang minimal 6 karakter
- Harus memiliki panjang maximal 12 karakter

Test Case Results:



5. NikValidationTest Case

File: NikValidationTest.java

```
6 package com.nusabank.helper;
7
8 import junitparams.FileParameters;
9 import junitparams.JUnit4ParamsRunner;
10 import static org.junit.Assert.assertEquals;
11 import org.junit.Test;
12 import org.junit.runner.RunWith;
13 /**
14  *
15  * @author User
16  */
17 @RunWith(JUnit4ParamsRunner.class)
18 public class NikValidationTest {
19     Validation validTest = new Validation();
20
21     @Test
22     @FileParameters("resources/data_nik.csv")
23     public void testIsValidNik(final String a, final boolean b) {
24         System.out.println("isValidNIK");
25         String val = a;
26         boolean expectedResult = b;
27         boolean result = validTest.isValidNIK(val);
28         assertEquals(expectedResult, result);
29     }
30 }
31
```

Test Case Results:



Terjadi kekeliruan regex pattern untuk validasi NIK, seharusnya NIK hanya memiliki nilai minimal 6 digit.

6. ContactValidationTest Case

File: ContactValidationTest.java

```
6 package com.nusabank.helper;
7
8 import junitparams.FileParameters;
9 import junitparams.JUnit4ParamsRunner;
10 import static org.junit.Assert.assertEquals;
11 import org.junit.Test;
12 import org.junit.runner.RunWith;
13 /**
14  *
15  * @author User
16  */
17 @RunWith(JUnit4ParamsRunner.class)
18 public class ContactValidationTest {
19     Validation validTest = new Validation();
20
21     @Test
22     @FileParameters("resources/data_contact.csv")
23     public void testIsValidContact(final String a, final boolean b) {
24         System.out.println("isValidContact");
25         String val = a;
26         boolean expectedResult = b;
27         boolean result = validTest.isValidPhoneNumber(val);
28         assertEquals(expectedResult, result);
29     }
30 }
```

Test Case Results:

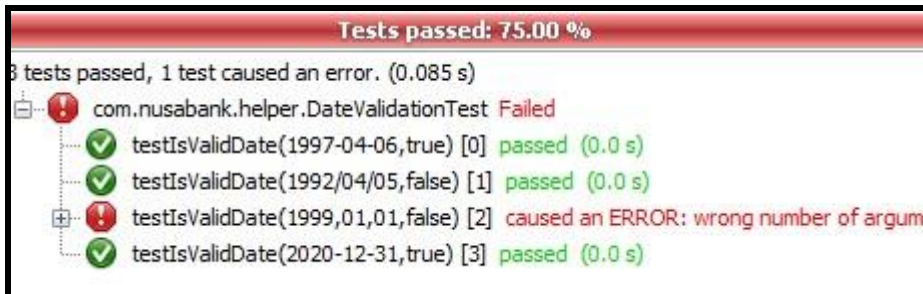


7. DateValidationTest Case

File: DateValidationTest.java

```
6   package com.nusabank.helper;
7
8   import junitparams.FileParameters;
9   import junitparams.JUnit4ParamsRunner;
10  import static org.junit.Assert.assertEquals;
11  import org.junit.Test;
12  import org.junit.runner.RunWith;
13
14  /**
15   * @author User
16   */
17  @RunWith(JUnit4ParamsRunner.class)
18  public class DateValidationTest {
19      Validation validTest = new Validation();
20
21      @Test
22      @FileParameters("resources/data_datetime.csv")
23      public void testIsValidDate(final String a, final boolean b) {
24          System.out.println("isValidDate");
25          String val = a;
26          boolean expResult = b;
27          boolean result = validTest.isValidDate(val);
28          assertEquals(expResult, result);
29      }
30  }
```

Test Case Results:



Terdapat error dimana 1999,01,01 tidak masuk kedalam rules regex pattern date.

8. PinValidationTest Case

File: PinValidationTest.java

```
6 package com.nusabank.helper;
7
8 import junitparams.FileParameters;
9 import junitparams.JUnit4ParamsRunner;
10 import static org.junit.Assert.assertEquals;
11 import org.junit.Test;
12 import org.junit.runner.RunWith;
13
14 /**
15  * @author User
16  */
17 @RunWith(JUnit4ParamsRunner.class)
18 public class PinValidationTest {
19     Validation validTest = new Validation();
20
21     @Test
22     @FileParameters("resources/data_pin.csv")
23     public void testIsValidPin(final String a, final boolean b) {
24         System.out.println("isValidPin");
25         String val = a;
26         boolean expResult = b;
27         boolean result = validTest.isValidPIN(val);
28         assertEquals(expResult, result);
29     }
30 }
```

Test Case Results:



Terjadi kekeliruan pada pattern regex untuk validasi PIN seharusnya PIN cukup 6 digit numeric saja.

9. FunctionTest Case

File Function.java pada project nusabank merupakan library helper yang berisi fungsi-fungsi bantuan untuk mempermudah proses development.

File: FunctionTest.java

```
46      @Before
47      public void setUp() {
48          vmn = new ViewManageNasabah();
49          vma = new ViewManageAdmin();
50          vMenuAdm = new ViewMenuAdmin();
51
52          mn = new ModelNasabah();
53          ma = new ModelAdmin();
54
55          mn.setUsername("dummy_nasabah");
56          ma.setUsername("dummy_admin");
57
58          vmn.setFileName("test.png");
59          vma.setFileName("test.png");
60      }
```

Initiate instance untuk dipakai pada unit test.

File: FunctionTest.java > testRandomCaptcha()

```
66      /**
67       * Test of randomCaptcha method, of class Function.
68       */
69      @Test
70      public void testRandomCaptcha() {
71          System.out.println("randomCaptcha");
72          int count = 5;
73          String expResult = "";
74          String result = Function.randomCaptcha(count);
75          assertNotNull("not null: " + result, result);
76      }
```

Test Case Random Captcha untuk mengetest apakah random captcha yang dihasilkan bernilai null atau tidak.

File: FunctionTest.java > testGenerateNoRek()

```
78  /**
79   * Test of generateNoRek method, of class Function.
80   */
81   @Test
82   public void testGenerateNoRek() {
83       System.out.println("generateNoRek");
84       String nmNasabah = "Tester";
85       String dob = "1990-02-02";
86       String dj = "2020-04-24";
87       Function instance = new Function();
88       String expResult = "";
89       String result = instance.generateNoRek(nmNasabah, dob, dj);
90       assertNotNull("No.Rekening isn't null", result);
91   }
```

File: FunctionTest.java > testUploadPhotoNasabah()

```
105  /**
106   * Test of uploadPhotoNasabah method, of class Function.
107   */
108   @Test
109   public void testUploadPhotoNasabah() {
110       System.out.println("uploadPhotoNasabah");
111       vrn = new ViewRegisNasabah();
112       vrn.setIdAdmin("1");
113       vrn.setFileName("test.png");
114       Function instance = new Function();
115       String expResult = "";
116       String result = instance.uploadPhotoNasabah(vrn, mn);
117       assertNotNull(expResult, result);
118   }
```

File: FunctionTest.java > testUploadPhotoAdmin()

```

120  /**
121   * Test of uploadPhotoAdmin method, of class Function.
122   */
123   @Test
124   public void testUploadPhotoAdmin() {
125       System.out.println("uploadPhotoAdmin");
126       Function instance = new Function();
127       String expResult = "";
128       String result = instance.uploadPhotoAdmin(vma, ma);
129       assertEquals(expResult, result);
130   }

```

File: FunctionTest.java > testUpdatePhotoNasabah()

```

93  /**
94   * Test of updatePhotoNasabah method, of class Function.
95   */
96   @Test
97   public void testUpdatePhotoNasabah() {
98       System.out.println("updatePhotoNasabah");
99       Function instance = new Function();
100       String expResult = "res/nasabah_photos/dummy_nasabah_NusaBank_null.png";
101       String result = instance.updatePhotoNasabah(vmn, mn);
102       assertEquals(expResult, result);
103   }

```

Test Case Results:

Tests passed: 100.00 %	
All 5 tests passed. (3.38 s)	generateNoRek
com.nusabank.helper.FunctionTest passed	-1793302305
testGenerateNoRek passed (2.24 s)	555703711
testRandomCaptcha passed (0.194 s)	131700378
testUpdatePhotoNasabah passed (0.131 s)	1793
testUploadPhotoAdmin passed (0.11 s)	5570
testUploadPhotoNasabah passed (0.631 s)	3170
	179355703170
	randomCaptcha
	updatePhotoNasabah
	res/nasabah_photos/dummy_nasabah_NusaBank_null.png
	uploadPhotoAdmin
	uploadPhotoNasabah
	D:\netbeans\workspace\NusaBank
	res/nasabah_photos/dummy_nasabah_NusaBank_null.png
	java.nio.file.NoSuchFileException: file...

10. CLoginNasabahTest Case

Mencoba testing controller untuk login nasabah apakah berfungsi sesuai dengan yang diinginkan atau tidak yakni apakah data username dan password yang di input user sama/matches dengan yang ada di database.

File: CLoginNasabahTest.java

```

33     @Before
34     public void setUp() {
35         login = new CLoginNasabah(new ViewLoginNasabah());
36     }
37     /**
38      * Test of doLogin method, of class CLoginNasabah.
39      * Test whether the username and password is matches
40      */
41     @Test
42     public void testDoLoginIsMatch() throws Exception {
43         String username = "asep1";
44         String password = "123456";
45         assertTrue("Login Matches!", login.doLogin(username, password));
46     }
47
48     /**
49      * Test of doLogin method, of class CLoginNasabah.
50      * Test whether the username and password isn't matches
51      */
52     @Test
53     public void testDoLoginIsNotMatch() throws Exception {
54         String username = "asepsooo";
55         String password = "123123";
56         assertFalse("Login Didn't Matches!", login.doLogin(username, password));
57     }
58 }
59
60

```

Saya coba membuat dua fungsi assertion dimana yang satu untuk mengetest jika data matching yang satunya untuk test apabila data tidak matches.

Test Case Results:

Tests passed: 100.00 %		res/nasabah_photos/asep1_NusaBank_2020-03-17_16-27.jpg
Both tests passed. (40.859 s)		
com.nusabank.controller.CLoginNasabahTest	passed	
testDoLoginIsMatch	passed (5.952 s)	
testDoLoginIsNotMatch	passed (34.838 s)	

Keduanya sukses, data pertama passed karena memang asep1 ada di database
Data kedua passed karena memang asepsooo tidak ada recordnya di database.