

Laporan Kualitas Perangkat Lunak

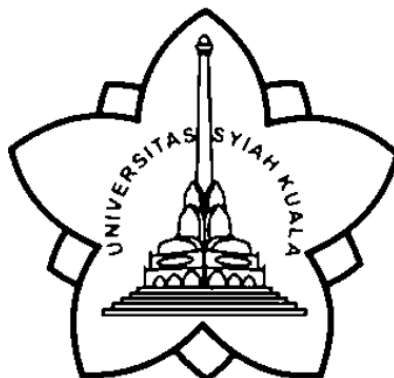
TUGAS JUNIT

disusun untuk memenuhi
tugas mata kuliah Kualitas Perangkat Lunak

oleh:

MUHAMMAD HABIL ASWAD

2208107010013



**JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH
2025**

DAFTAR ISI

SOAL	2
JAWAB.....	3
1. Membuat Proyek baru Android sederhana	3
2. Buat minimal 2 instrumental test dan minimal 2 local test	3
3. Jelaskan test case yang dibuat berupa comment.....	6
4. Upload project pada github.	6
5. Sertakan link github pada laporan tugas ini	7

SOAL

Simak video materi pada bagian JUnit pada Android. Kerjakan tugas berikut:

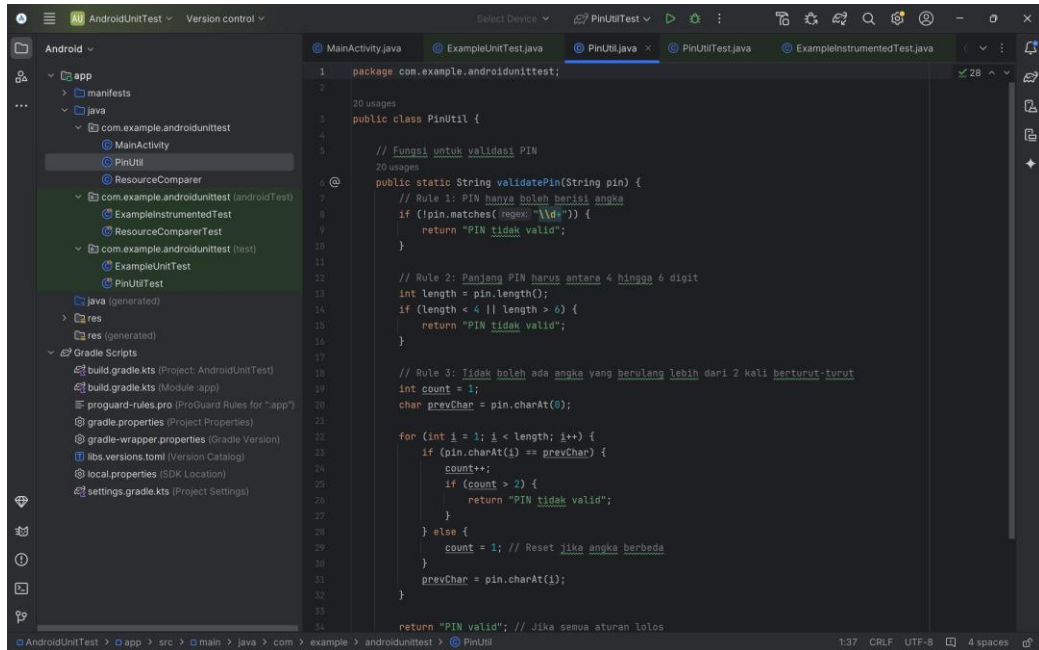
1. Buat sebuah projek baru android sederhana.
2. Buat minimal 2 instrumental test dan minimal 2 local test
3. Jelaskan [test case](#) yang dibuat berupa comment `/* */` pada class unit test yang dibuat. Jika tidak menjelaskan pada kode maka nilai akan dikurangi 50
4. Upload project anda pada github.
5. Sertakan link github pada laporan tugas ini
6. Kumpulkan laporan project pada tugas ini berupa pdf

Due date: 10 Maret 2025, pk. 16.00 sore. (Dipresentasikan dan dibahas pada 10 Maret 2025)

JAWAB

1. Membuat Proyek baru Android sederhana

Saya telah membuat proyek baru Android sederhana bernama AndoridUnitTest berupa hal yang sama seperti yang saya buat pada Tugas Pengujian Perangkat Lunak, yaitu code untuk melakukan validasi PIN. Namun kali ini menggunakan bahasa pemrograman Java dan menggunakan Android Studio Code seperti dapat dilihat pada **Gambar 1**.



Gambar 1. Tampilan Proyek baru Android

2. Buat minimal 2 instrumental test dan minimal 2 local test

Untuk **instrumental test** (frontend) saya membuat file ResourceComparerTest.java hasil generate test dari ResourceComparer.java yang terdapat 2 fungsi/method di dalamnya, yaitu fungsi untuk menguji apakah teks yang diambil dari resource akan sama dengan yang diinginkan (**Gambar 2**), teks tidak sesuai dengan yang diinginkan (**Gambar 3**), teks sama namun dengan style yang berbeda atau mengabaikan case sensitive (**Gambar 4**), dan teks yang benar-benar salah walaupun mengabaikan case sensitive (**Gambar 5**).

```
@Test
public void testResourceMatchesCorrectString() {
    String expectedText = "Hello World"; // Teks yang seharusnya ada di strings.xml
    assertTrue(ResourceComparer.isEqual(context, R.string.hello_world, expectedText));
}
```

Gambar 2. Method mengecek kesesuaian teks

```
@Test
public void testResourceDoesNotMatchIncorrectString() {
    String wrongText = "Hello"; // Teks yang berbeda dari `strings.xml`
    assertFalse(ResourceComparer.isEqual(context, R.string.hello_world, wrongText));
}
```

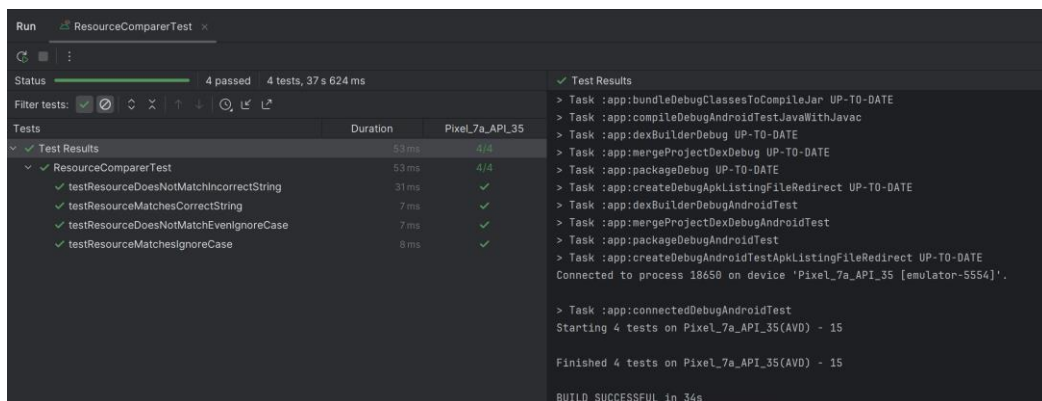
Gambar 3. Method mengecek ketidaksesuaian teks

```
@Test
public void testResourceMatchesIgnoreCase() {
    String differentCaseText = "hello world"; // Huruf kecil, tetapi seharusnya masih cocok
    assertTrue(ResourceComparer.isEqualIgnoreCase(context, R.string.hello_world, differentCaseText));
}
```

Gambar 4. Method mengecek kesesuaian teks tanpa memerhatikan case sensitive

```
@Test
public void testResourceDoesNotMatchEvenIgnoreCase() {
    String completelyDifferentText = "Hi Universe"; // Berbeda secara keseluruhan
    assertFalse(ResourceComparer.isEqualIgnoreCase(context, R.string.hello_world, completelyDifferentText));
}
```

Gambar 5. Method mengecek teks yang salah walaupun mengabaikan case sensitive



Gambar 6. Hasil uji Class ResrouceComparerTest

Untuk **local test** (backend), saya memilih untuk melakukan pengecekan validasi PIN dari pengguna dengan beberapa rule yang diterapkan. Saya membuat file `PinUtilTest.java`, hasil generate test dari `PinUtil.java` yang isinya berupa pengujian-pengujian untuk PIN yang sesuai dengan rule yang diterapkan. Saya membuat 2 Method, yaitu method `testValidPIN` (**Gambar 7**) untuk mengecek kombinasi PIN yang seharusnya sesuai dengan ketentuan, dan method `testInvalidPin` (**Gambar 8**) untuk mengecek beberapa PIN yang tidak sesuai rule.

```

@Test
public void testValidPin() {
    assertEquals( expected: "PIN valid", PinUtil.validatePin("1234")); //
    assertEquals( expected: "PIN valid", PinUtil.validatePin("567890"));
    assertEquals( expected: "PIN valid", PinUtil.validatePin("998877"));
    assertEquals( expected: "PIN valid", PinUtil.validatePin("1100")); //
    assertEquals( expected: "PIN valid", PinUtil.validatePin("135792"));
    assertEquals( expected: "PIN valid", PinUtil.validatePin("2468")); //
    assertEquals( expected: "PIN valid", PinUtil.validatePin("99123")); //
    assertEquals( expected: "PIN valid", PinUtil.validatePin("808080"));
    assertEquals( expected: "PIN valid", PinUtil.validatePin("121314"));
    assertEquals( expected: "PIN valid", PinUtil.validatePin("4321")); //
}

```

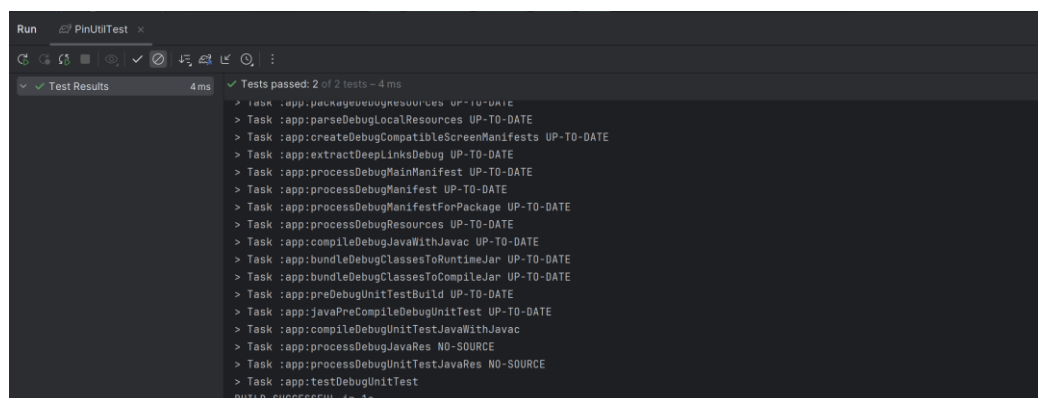
Gambar 7. Method testValidPin

```

@Test
public void testInvalidPin() {
    assertEquals( expected: "PIN tidak valid", PinUtil.validatePin("abcd")); //
    assertEquals( expected: "PIN tidak valid", PinUtil.validatePin("1112")); //
    assertEquals( expected: "PIN tidak valid", PinUtil.validatePin("123")); //
    assertEquals( expected: "PIN tidak valid", PinUtil.validatePin("1234567"));
    assertEquals( expected: "PIN tidak valid", PinUtil.validatePin("12@34")); //
    assertEquals( expected: "PIN tidak valid", PinUtil.validatePin("999999"));
    assertEquals( expected: "PIN tidak valid", PinUtil.validatePin("122333"));
    assertEquals( expected: "PIN tidak valid", PinUtil.validatePin("a1b2c3"));
    assertEquals( expected: "PIN tidak valid", PinUtil.validatePin("888000"));
    assertEquals( expected: "PIN tidak valid", PinUtil.validatePin("")); // Str
}

```

Gambar 8. Method testInvalidPin



Gambar 8. Hasil Uji class PinUtilTest

3. Jelaskan test case yang dibuat berupa comment `/** */` pada class unit test yang dibuat. Jika tidak menjelaskan pada kode maka nilai akan dikurangi 50

Penjelasan inshaallah sudah saya buat sejelas-jelasnya melalui komen di atas setiap code method, dan di samping masing-masing percobaan atau test case nya, seperti dapat dilihat pada **Gambar 9** dan **Gambar 10**.

```
/**
 * Menguji berbagai PIN yang memenuhi aturan validasi.
 *
 * Test case ini memastikan bahwa:
 * 1. PIN terdiri dari **4 hingga 6 digit angka**.
 * 2. Tidak mengandung karakter selain angka.
 * 3. Tidak memiliki angka yang berulang lebih dari **2 kali berturut-turut**.
 *
 * Data uji yang digunakan:
 * - "1234", 4 digit, angka semua, valid.
 * - "567890", 6 digit, angka semua, valid.
 * - "998877", 6 digit, angka tidak berulang lebih dari 2 kali berturut-turut, valid.
 * - "0011", 4 digit angka yang sama, tapi masih valid karena tidak lebih dari 2 kali berturut-turut.
 * - "135792", 6 digit, kombinasi angka unik, valid.
 * - "2468", 4 digit, semua angka berbeda, valid.
 * - "99123", 6 digit, '99' muncul 2 kali, valid.
 * - "808080", 6 digit, pola berulang tetapi tidak lebih dari 2 kali berturut-turut, valid.
 * - "121314", 6 digit, angka unik, valid.
 * - "4321", 4 digit angka dalam pola menurun, valid.
 */
@Test
public void testValidPin() {
```

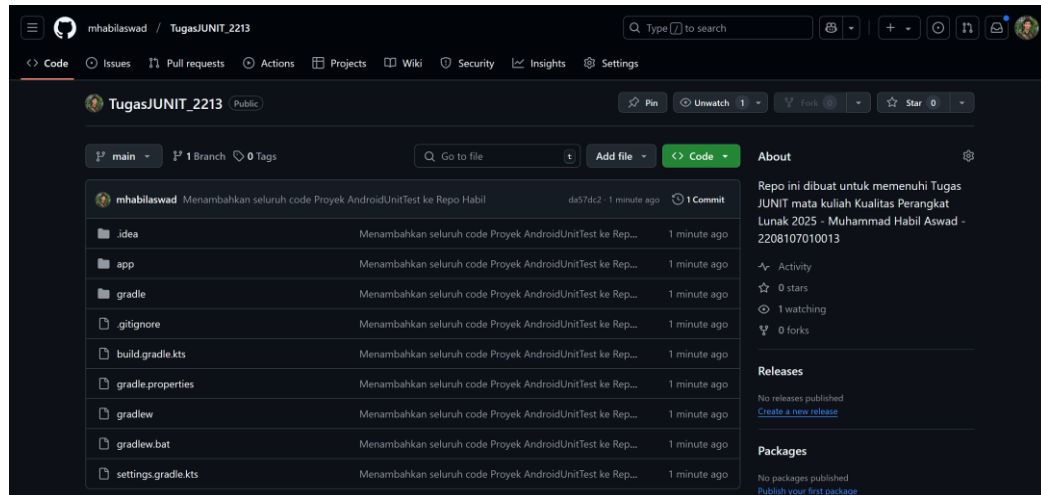
Gambar 9. Komen penjelasan di atas setiap method testing

```
() {
    "PIN valid", PinUtil.validatePin("1234")); // PIN 4 digit, valid
    "PIN valid", PinUtil.validatePin("567890")); // PIN 6 digit, valid
    "PIN valid", PinUtil.validatePin("998877")); // Tidak ada angka yang berulang >2 kali
    "PIN valid", PinUtil.validatePin("1100")); // Semua angka sama tapi tidak lebih dari 2 kali berturut-turut
    "PIN valid", PinUtil.validatePin("135792")); // Tidak ada angka yang berulang berturut-turut
    "PIN valid", PinUtil.validatePin("2468")); // PIN dengan 4 digit acak, valid
    "PIN valid", PinUtil.validatePin("99123")); // 99 diulang 2x, valid
    "PIN valid", PinUtil.validatePin("808080")); // Tidak ada angka yang berulang lebih dari 2 kali berturut-turut
    "PIN valid", PinUtil.validatePin("121314")); // Kombinasi angka yang valid
    "PIN valid", PinUtil.validatePin("4321")); // PIN menurun tanpa pengulangan berlebihan
}
```

Gambar 10. Komen penjelasan di samping setiap test case

4. Upload project pada github.

Proyek AndroidUnitTest sudah saya tambahkan ke repository github bernama TugasJUNIT_2213 seperti pada **Gambar 11**.



Gambar 11. Repository TugasJUNIT_2213

5. Sertakan link github pada laporan tugas ini

Repository GitHub **TugasJUNIT_2213** dapat diakses melalui link berikut ini:

https://github.com/mhabilaswad/TugasJUNIT_2213