

**MAKALAH**  
**METODE PENGUJIAN WHITE BOX DAN BLACK BOX**



DISUSUN OLEH

R. Ayu Setia Anggini

NPM : 2010010345

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ISLAM KALIMANTAN**  
**MUHAMMAD ARSYAD AL BANJARI**  
**BANJARMASIN**

## **KATA PENGANTAR**

Rasa syukur saya haturkan kepada Tuhan Yang Maha Kuasa, karena berkat karunianya saya dapat menyusun makalah ini dengan baik dan selesai tepat pada waktunya.

Penyusunan makalah ini bertujuan untuk memenuhi tugas mata kuliah Testing dan Implementasi di Bidang TI. Selain itu, makalah ini juga bertujuan untuk memberikan tambahan wawasan bagi saya sebagai penulis dan bagi para pembaca.

Saya selaku penulis tidak lupa untuk mengucapkan terima kasih kepada Bapak Yusup Indra Wijaya, S.Kom., M.Kom. selaku dosen pengampu mata kuliah Testing dan Implementasi di Bidang TI ini. Tidak lupa bagi pihak-pihak lain yang telah mendukung penulisan makalah ini saya juga mengucapkan terima kasih.

Terakhir, saya menyadari bahwa makalah ini masih jauh dari kesempurnaan. Maka dari itu saya membutuhkan kritik dan saran yang bisa membangun agar kedepannya bisa menulis makalah dengan lebih baik lagi. Semoga makalah ini bermanfaat bagi para pembaca, dan bagi saya khususnya sebagai penulis.

Banjarmasin, 17 April 2023

R. Ayu Setia Anggini

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
A. Latar Belakang .....	1
B. Rumusan Masalah .....	2
C. Tujuan .....	2
<b>BAB II PEMBAHASAN.....</b>	<b>3</b>
1. Metode Pengujian White Box .....	3
2. Metode Pengujian Black Box .....	7
<b>BAB III PENUTUP.....</b>	<b>12</b>
A. Kesimpulan .....	12
<b>DAFTAR PUSTAKA .....</b>	<b>iii</b>

## **BAB I**

### **PENDAHULUAN**

#### **A. Latar Belakang**

Testing atau pengujian perangkat lunak adalah suatu proses yang dibuat sedemikian rupa dalam mengidentifikasi ketidaksesuaian terhadap kinerja dari suatu program atau sistem untuk mendapatkan hasil yang diharapkan (Novianti, 2014). Berbagai macam metode dilakukan dalam pengujian dengan mengoprasikan perangkat lunak dalam suatu kondisi yang dikendalikan, seperti halnya dengan melakukan pengecekan pada sistem apakah sistem yang dibangun telah sesuai sebagaimana yang telah ditetapkan menurut spesifikasi, kemudian mendeteksi error atau menemukan bugs yang ada pada sistem, dan melakukan validasi apakah sistem sudah siap untuk digunakan oleh pengguna atau belum. Di samping itu pengujian juga berguna dalam melakukan evaluasi efektifitas kerja bagi pengembang perangkat lunak ataupun penguji perangkat lunak.

Pengujian memiliki peranan penting dalam suatu kontruksi perangkat lunak, sehingga perencanaan pengujian yang baik sangat perlu diperhatikan. Dalam teori pengujian perangkat lunak, terdapat beberapa rancangan metode pengujian perangkat lunak salah satunya adalah white box testing. Banyak pengembang perangkat lunak biasanya langsung melakukan pengujian secara usability dan fungsionalitas seperti salah satu contohnya yaitu pengujian pada tampilan antarmuka sistem atau biasa disebut dengan black box testing.

## B. Rumusan Masalah

Adapun rumusan masalah dalam makalah ini yaitu sebagai berikut :

- 1) Bagaimana metode pengujian white box?
- 2) Bagaimana metode pengujian black box?

## C. Tujuan

Adapun tujuan dari penulisan makalah ini yaitu untuk mengetahui bagaimana metode pengujian white box dan metode pengujian black box beserta contohnya.

## **BAB II PEMBAHASAN**

### **1. Metode Pengujian White Box**

White box testing adalah salah satu teknik pengujian software yang berfokus pada komponen di dalam software seperti desain, struktur coding-an, dan cara kerja software dari awal hingga akhir. Tujuan dari white box testing adalah untuk mengetahui alur pengiriman data dari input hingga output, kegunaan, dan keamanan software. Pada white box testing, kode yang digunakan untuk membuat software bisa dilihat oleh penguji. Karena itu, white box testing disebut juga dengan clear box testing, open box testing, code-based testing, atau structural testing. White box testing harus dilakukan oleh profesional yang memahami kode dengan baik. Setelah itu, tester harus menulis beberapa kode untuk dimasukkan dalam uji kasus (test case) dan menjalankannya.

Ada beberapa tipe white box testing, dan setiap jenisnya memiliki teknik pengujian yang berbeda. Di antara teknik white box testing, berikut tiga teknik yang paling utama:

#### **a) Statement Coverage**

Dalam bahasa pemrograman, pernyataan (statement) adalah baris kode atau instruksi agar komputer dapat memahami dan bertindak sesuai kode tersebut. Statement coverage berfungsi menguji setiap kemungkinan pernyataan tersebut setidaknya sekali untuk menemukan kode yang salah. Namun sebelumnya, pernyataan harus dikompilasi dan diubah menjadi kode objek terlebih dahulu, dan melakukan tindakan saat software dijalankan.

#### **b) Branch Coverage**

Branch dalam pemrograman seperti pernyataan IF (IF statement) dengan dua opsi jawaban, yaitu benar dan salah. Teknik ini memeriksa apakah setiap cabang atau jalur dijalankan setidaknya sekali selama software digunakan.

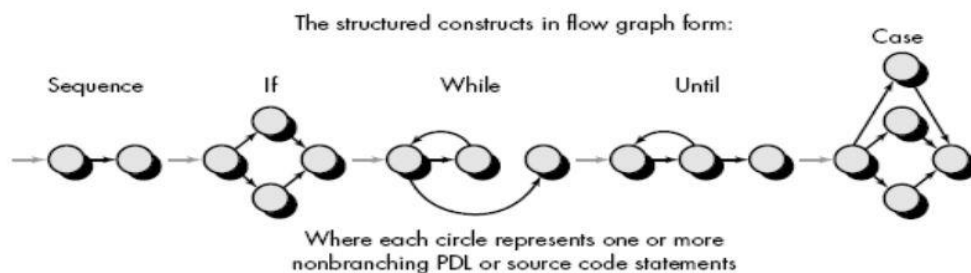
c) Path Coverage

Path coverage merupakan teknik pengujian komprehensif di mana seluruh kemungkinan jalur dalam program diuji untuk memeriksa jalur mana saja yang dilewati pada status awal program dijalankan, status menengah, dan status akhir. Teknik path coverage lebih menyeluruh dibandingkan dengan branch coverage, sehingga teknik ini umumnya digunakan untuk menguji program yang kompleks.

Teknik white box testing lain yang sering digunakan antara lain:

a) Basis Path Testing

Metode ini memungkinkan perancang kasus uji untuk mengukur kompleksitas logis dari rancangan prosedural dan menggunakannya sebagai pedoman untuk menetapkan sekumpulan jalur eksekusi dasar (basis set). Kasus uji yang dilakukan untuk menggunakan basis set tersebut dijamin untuk menggunakan setiap statement dalam program paling tidak sekali selama pengujian. Metode ini menggunakan notasi flow graph yang menggambarkan aliran kontrol logika yang menggunakan notasi yang ditunjukkan pada gambar di bawah ini:



b) Loop Testing

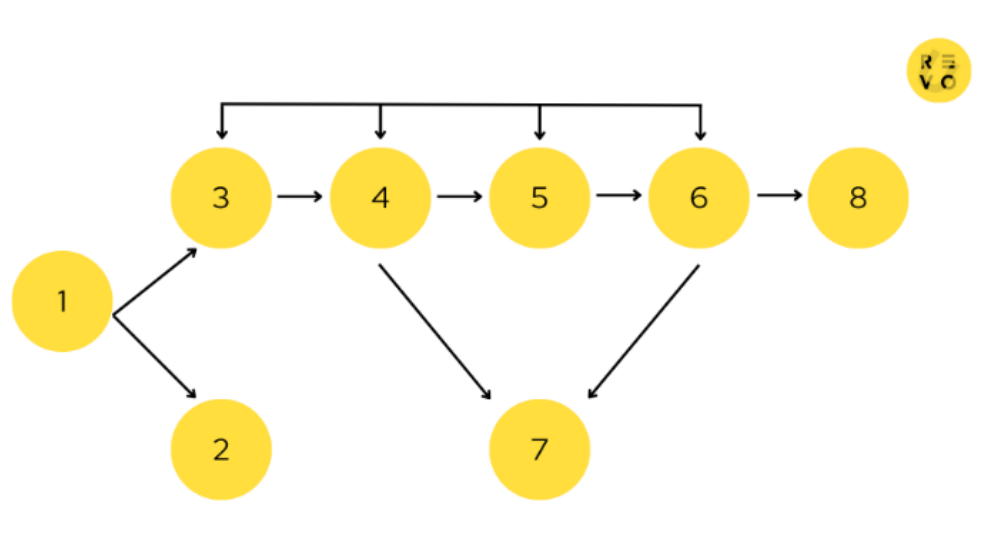
Pengujian berulang dengan tiga strategi berbeda, yaitu single loop, concatenated loop, dan nested loop.

c) Compound Condition Coverage

Menguji setiap kondisi dengan beberapa jalur dan kombinasi jalur yang berbeda.

Contoh white box testing:

Untuk melakukan white box testing dengan teknik path coverage, tester harus menggambar diagram (flowchart) kontrol kode untuk melihat kemungkinan jalur yang akan dilewati saat program dijalankan. Contoh kode dan jalurnya bisa dilihat pada gambar di bawah ini:



Maka berdasarkan diagram di atas, kemungkinan jalur yang dilewati antara lain:

- 1, 2
- 1, 3, 4, 5, 6, 8
- 1, 3, 4, 5, 6, 7
- 1, 3, 4, 7, 6, 8, dan sebagainya

Karena ada banyaknya jalur yang dihasilkan, maka kemungkinan terdapat jalur yang tidak efisien, berulang, atau salah.

Dengan melakukan teknik path coverage, kode yang menghasilkan jalur yang tak efisien akan dimodifikasi.



White box testing menguji seluruh komponen dalam suatu program atau aplikasi, sehingga memiliki keuntungan sebagai berikut:

- a) Dapat menemukan bug atau error yang tersembunyi, termasuk kesalahan pada tipografi dan sintaks.
- b) Otomatisasi dan optimasi kode menjadi mudah.
- c) Pengujian bisa dilakukan pada tahap awal tanpa perlu antarmuka seperti pada black box testing.
- d) Mengoptimalkan kesalahan penghapusan kode dan membantu dalam menghapus baris kode tambahan.

Namun, white box testing juga memiliki kekurangan antara lain:

- a) Pengujian sangat rumit, butuh waktu yang lama, dan biayanya sangat mahal.
- b) Tester harus memiliki pemahaman programming dan coding yang mendalam.
- c) Ada kemungkinan tester terlalu fokus pada cara kerja internal software dan melewatkan masalah eksternal.
- d) Ada kemungkinan tester memiliki pandangan yang bias terhadap software karena mereka terbiasa dengan cara kerja internalnya, sehingga kesalahan dalam produksi bisa lebih banyak.
- e) Desain ulang dan penulisan ulang kode memerlukan test case untuk bisa ditulis lagi.
- f) Fungsionalitas yang hilang tidak dapat dideteksi saat kode diuji.

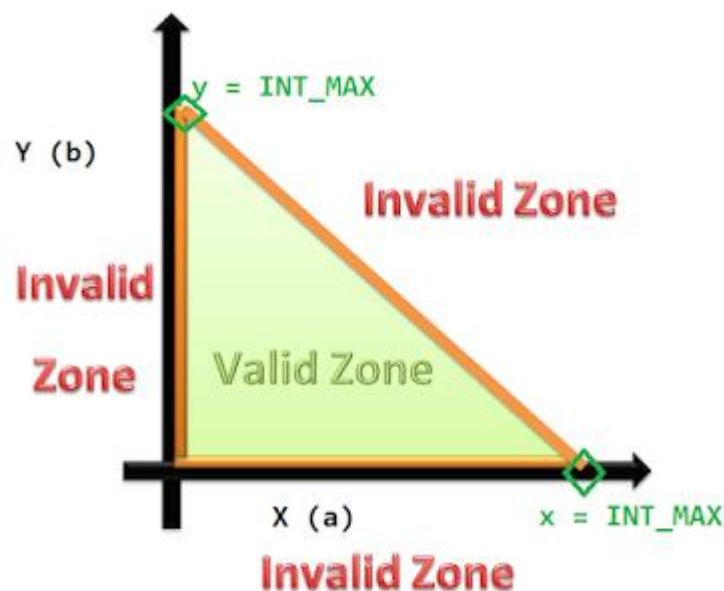
## 2. Metode Pengujian Black Box

Black box testing adalah proses menguji fungsi perangkat lunak (software) atau aplikasi dari sudut pandang pengguna, tanpa mengetahui struktur internal atau desain struktur tersebut. Sederhananya, black box testing hanya melakukan penilaian dari apakah sistem bisa memberikan output atau hasil sesuai dengan input (informasi atau instruksi yang diterima sistem). Jika hasilnya sesuai, maka sistem dinyatakan bisa berfungsi dengan baik. Jika sistem gagal menjalankan prosedur yang diminta, maka dinyatakan butuh perbaikan. Karena itu black box testing juga disebut dengan behavioral testing, closed-box, atau specification-based testing.

Dirangkum dari Javatpoint, berikut teknik black box testing yang digunakan untuk menguji perangkat lunak:

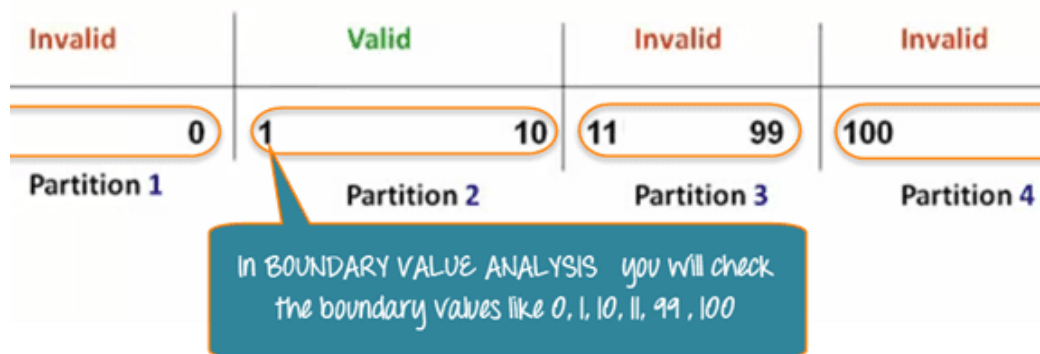
### a) Equivalence Partitioning Technique

Teknik ini membagi data input menjadi dua bagian, yaitu nilai valid dan invalid, dan dilakukan pengujian secara terpisah. Kedua bagian tersebut harus menunjukkan perilaku yang sama untuk dinyatakan pengujian berhasil.



b) Boundary Value Technique

Teknik boundary value digunakan untuk menguji error pada nilai batas atas dan bawah suatu variabel. Misalnya, variabel usia memiliki batas bawah (minimum) yaitu 18 tahun, dan batas atasnya (usia maksimum) adalah 30 tahun. Teknik ini digunakan karena banyak aplikasi yang memiliki masalah pada nilai batas, terutama batas atas.



c) All-Pair Testing Technique

Teknik ini menguji seluruh kemungkinan kombinasi data diskrit. Data diskrit adalah data dengan nilai yang pasti dan terbatas, biasanya dalam bentuk bilangan bulat atau ganjil. Misalnya, 2 kucing, 1 laptop, 5 baju. Metode kombinasi ini digunakan untuk menguji aplikasi yang menggunakan input kotak centang, input tombol radio, kotak daftar, kotak teks, dan lain-lain.

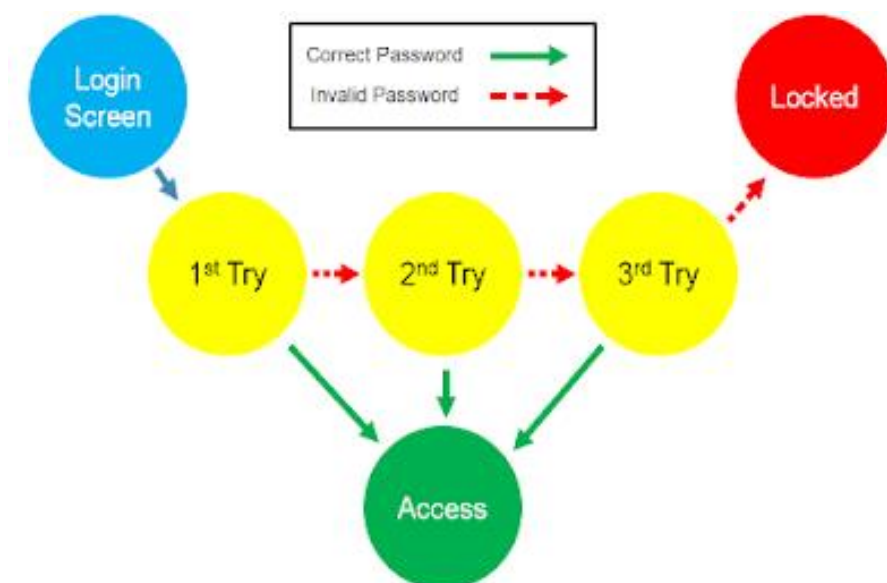
d) Decision Tabel Technique

Teknik ini dilakukan dengan pendekatan sistematis, di mana kombinasi input dirangkum dalam sebuah tabel. Decision table technique cocok untuk menguji fungsi yang memiliki hubungan logis antara dua atau lebih input. Sebagai contoh, Andin ingin log in ke Facebook. Sebelumnya, ia diminta untuk memasukkan e-mail dan password yang telah disimpan oleh Facebook. Jika Andin memasukkan e-mail dan password yang benar, maka ia diarahkan ke halaman Beranda Facebook-nya. Tapi jika Andin memasukkan e-mail dan password yang salah, ia akan tetap di halaman log in.

	Rule 1	Rule 2	Rule 3	Rule 4
<b>Condition</b>				
End of month	No	Yes	Yes	Yes
Salary Transferred	N/A	No	Yes	Yes
Provident fund	N/A	N/A	No	Yes
<b>Action</b>				
Income tax	No	No	Yes	Yes
Provident fund	No	No	No	Yes

e) State Transition Technique

Teknik state transition digunakan untuk mengetahui apakah fungsi software bisa bekerja jika mendapatkan nilai input yang berbeda. Hal ini karena keadaan output sistem bisa berubah tergantung pada kondisi atau peristiwa. State transition technique digunakan pada jenis aplikasi yang memberikan jumlah percobaan tertentu untuk mengakses aplikasi. Misalnya seperti pada ATM, jika user memasukkan PIN yang salah berkali-kali maka output-nya adalah PIN terblokir dengan sendirinya.



f) Cause-Effect Technique

Cause-effect technique menggunakan grafik untuk menggambarkan hubungan antara penyebab error dan efeknya.

g) Error Guessing Technique

Pada teknik ini, identifikasi error pada aplikasi didasarkan pada pengalaman dan pengetahuan penguji.

h) Use Case Technique

Teknik ini menguji setiap fungsi software dengan cara menjalankan sistem dari tahap awal hingga akhir.

Contoh black box testing:

Salah satu contoh pengujian black box adalah kasus Andin yang telah disebutkan di atas. Misalnya, suatu hari Andin ingin log in ke akun Facebook-nya. Ia memasukkan e-mail yang telah teregistrasi dan password yang diatur sebelumnya. Namun ternyata, ada pemberitahuan bahwa password yang dimasukkan oleh Andin salah. Pada halaman log in, Facebook memberi pilihan apakah ia ingin me-reset passwordnya atau log in langsung melalui e-mail. Andin mencoba memasukkan password yang berbeda, dan ternyata berhasil. Ia pun diarahkan ke halaman Beranda akunnya. Contoh di atas menggunakan teknik decision table karena terdapat dua input atau kondisi berbeda (e-mail dan password) yang saling berhubungan, dan output-nya tergantung pada kesesuaian hubungan tersebut.

**Ketentuan:**

- Jika e-mail dan password yang dimasukkan benar, maka diarahkan ke Beranda Facebook
- Jika e-mail atau password yang dimasukkan salah, maka akan tetap di halaman *log in*
- Jika e-mail dan password yang dimasukkan salah, maka akan tetap di halaman *log in*

<i>E-mail</i>	Benar	Benar
<i>Password</i>	Salah	Benar
<b>Hasil</b>	<i>E-mail</i> benar tapi <i>password</i> salah, tetap di halaman <i>log in</i>	<i>E-mail</i> dan <i>password</i> benar, diarahkan ke halaman Beranda

Black box testing adalah tes yang khusus dirancang untuk mencari error sistem dari sudut pandang pengguna. Siapapun bisa melakukan black box testing, tidak harus orang dengan latar belakang programming.

Keuntungan lain dari black box testing meliputi:

- a) Pengujian dapat langsung dilakukan setelah software dikembangkan.
- b) Black box testing efektif untuk menguji aplikasi yang besar dan kompleks.
- c) Error dan inkonsistensi sistem bisa dideteksi pada awal pengujian.

Terlepas dari manfaatnya, black box testing memiliki kekurangan antara lain:

- a) Jika penguji tak memiliki latar belakang programming, ada kemungkinan penguji mengabaikan kondisi atau masalah yang seharusnya ingin diujikan.
- b) Black box testing tidak bisa melakukan complete test coverage, yaitu pengujian yang melibatkan kode yang digunakan.
- c) Ada kemungkinan black box testing tidak menguji kemungkinan input yang bisa terjadi dan output-nya.

### **BAB III**

### **PENUTUP**

#### **A. Kesimpulan**

Pengujian memiliki peranan penting dalam suatu konstruksi perangkat lunak, sehingga perencanaan pengujian yang baik sangat perlu diperhatikan. Dalam teori pengujian perangkat lunak, terdapat beberapa rancangan metode pengujian perangkat lunak salah satunya adalah white box testing. White box testing harus dilakukan oleh profesional yang memahami kode dengan baik. Setelah itu, tester harus menulis beberapa kode untuk dimasukkan dalam uji kasus (test case) dan menjalankannya. Banyak pengembang perangkat lunak biasanya langsung melakukan pengujian secara usability dan fungsionalitas seperti salah satu contohnya yaitu pengujian pada tampilan antarmuka sistem atau biasa disebut dengan black box testing. Sederhananya, black box testing hanya melakukan penilaian dari apakah sistem bisa memberikan output atau hasil sesuai dengan input (informasi atau instruksi yang diterima sistem). Jika hasilnya sesuai, maka sistem dinyatakan bisa berfungsi dengan baik. Jika sistem gagal menjalankan prosedur yang diminta, maka dinyatakan butuh perbaikan. Karena itu black box testing juga disebut dengan behavioral testing, closed-box, atau specification-based testing.

## **DAFTAR PUSTAKA**

[Apa itu White Box Testing? Pengertian dan contoh 2023 | RevoU](#)

[Apa itu Black Box Testing? Pengertian dan contoh 2023 | RevoU](#)

[Teknik Pengujian Black-box Testing dan White-box Testing | S1 Sistem Komputer S.Kom \(stekom.ac.id\)](#)