

# **PENGUJIAN PERANGKAT LUNAK DENGAN MENGGUNAKAN METODE WHITE BOX DAN BLACK BOX**

**Abdul Rouf**

Sistem Informasi – STMIK HIMSYA Semarang

Email: roufclass@gmail.com

---

## ***Abstrak***

*Pengujian adalah proses untuk menemukan error pada perangkat lunak sebelum dikirim kepada pengguna. Pengujian Software adalah kegiatan yang ditujukan untuk mengevaluasi atribut atau kemampuan program dan memastikan bahwa itu memenuhi hasil yang dicari, atau suatu investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji (under test), Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat risiko pada implementasinya.*

**Kata Kunci:** Perangkat Lunak

---

## **1.1 Latar Belakang**

Pengujian perangkat lunak adalah sangat diperlukan dalam suatu system informasi, dimana dengan melakukan suatu pengujian akan ditemukan kesalahan atau error yang muncul dari system perangkat lunak tersebut. Dengan demikian tentunya seorang programmer akan bisa mengetahui dan apa yang harus dikerjakan selanjutnya.

Pentingnya pengujian perangkat lunak dan implikasinya yang mengacu pada kualitas perangkat lunak tidak dapat terlalu ditekan karena melibatkan sederetan aktivitas produksi di mana peluang terjadinya kesalahan manusia sangat besar dan arena ketidakmampuan manusia untuk melakukan dan berkomunikasi dengan sempurna maka pengembangan perangkat lunak diiringi dengan aktivitas jaminan kualitas

## **1.2 Tujuan Pengujian**

Tujuan dalam pengujian dari sebuah perangkat lunak adalah sebagai berikut :

1. Proses menjalankan program dengan maksud mencari kesalahan (error)
2. Kasus uji yang baik adalah kasus yang memiliki peluang untuk mendapatkan kesalahan yang belum diketahui
3. Pengujian dikatakan berhasil bila dapat memunculkan kesalahan yang belum diketahui

4. Jadi pengujian yang baik bukan untuk memastikan tidak ada kesalahan tetapi untuk mencari sebanyak mungkin kesalahan yang ada pada program

### 1.3 Metode Penelitian

Metode penelitian yang dilakukan oleh peneliti adalah dengan eksperimen dari metode pengujian tersebut terhadap sebuah perangkat lunak.

### 1.4 Pembahasan

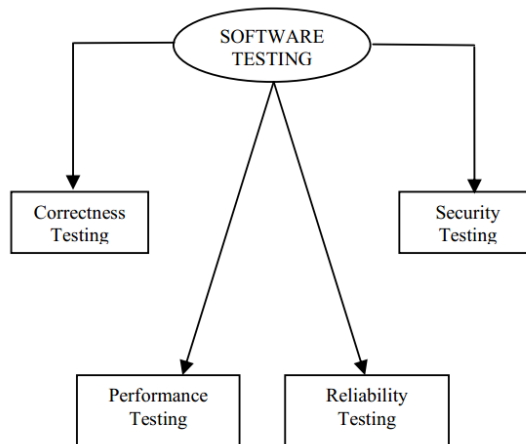
Pengujian perangkat lunak seharusnya menghabiskan waktu 30% – 40% dari total biaya pembangunan perangkat lunak. Pengujian merupakan bagian dari salah satu tugas software verification dan validation, yang merupakan bagian dari *software quality assurance*.

Pengujian perangkat lunak mencakup :

- a) Strategi : Mengintegrasikan metode perancangan kasus uji dalam sekumpulan langkah yang direncanakan
- b) metode pengujian, mencakup Perancangan kasus uji dengan menggunakan metode White Box atau Black Box

#### Siapa saja yang berhubungan dengan pengujian ?

- a) Customer, tim yang mengontrak pengembang untuk mengembangkan perangkat lunak.
- b) Pengguna , kelompok yang akan menggunakan perangkat lunak
- c) Pengembang perangkat lunak, tim yang membangun perangkat lunak
- d) Tim Pengujian perangkat lunak, tim khusus yang bertugas untuk menguji fungsi-fungsi pada perangkat lunak.



Gb.1 Teknik Pengujian Software

### **Prinsip Pengujian**

- a) Semua pengujian harus dapat dirunut sampai kepada spesifikasi kebutuhan perangkat lunak.
- b) Pengujian harus dimulai dari lingkup yang kecil ke lingkup yang besar
- c) Pengujian yang mendalam tidak mungkin dilakukan karena tidak mungkin mengeksekusi semua jalur permutasi
- d) Supaya efektif (memiliki probabilitas yang tinggi dalam menemukan kesalahan), Pengujian harus dilakukan oleh pihak lain yang independen
- e) Pengujian harus direncanakan jauh sebelum dilakukan

### **Kualitas Pengujian yang baik**

- a) mencakup semua kemungkinan skenario pengoperasian perangkat lunak
- b) mencakup sebanyak mungkin jalur yang dibentuk dari struktur program
- c) tidak terlalu sederhana dan tidak terlalu rumit

### **Kasus Uji (Test Case)**

Kasus uji yang baik :

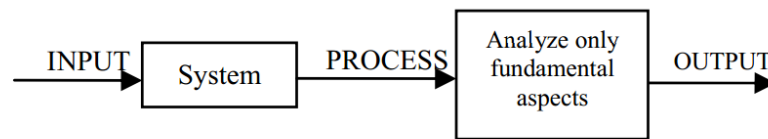
Adalah kasus uji yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya. Perancangan test case adalah perancangan untuk menyediakan kemungkinan-kemungkinan yang cukup tinggi untuk menemukan kesalahan (sesuai dengan tujuan Uji coba) dengan jumlah waktu dan usaha yang minimum.

Sebelum melakukan pengujian perlu dipersiapkan Test Case. Test Case terlebih dahulu agar diperoleh terlebih dahulu agar diperoleh kemungkinan tertinggi dalam menemukan kemungkinan tertinggi dalam menemukan kesalahan dengan waktu dan usaha yang minimum. Desain Test Case dapat dilakukan minimum. Desain Test Case dapat dilakukan melalui berbagai teknik pengujian diantaranya adalah :

- a) Black Box : pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan
- b) White Box, pengujian untuk memperlihatkan cara kerja dari produk secara rinci sesuai dengan spesifikasinya

### **Pengujian Black Box**

- a) Metode Black Box memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.
- b) Black Box dapat menemukan kesalahan dalam kategori berikut :
  1. Fungsi-fungsi yang tidak benar atau hilang
  2. Kesalahan interface
  3. Kesalahan dalam struktur data atau akses basisdata eksternal
  4. Inisialisasi dan kesalahan terminasi
  5. validitas fungsional
  6. kesensitifan sistem terhadap nilai input tertentu
  7. batasan dari suatu data



Gb.2. Sistem kerja dari Teknik Pengujian Black Box

Tipe dari Black Box Testing :

1. Equivalence class partitioning
2. Sample testing
3. Limit testing
4. Robustness testing
5. Behavior testing
6. Requirement testing

Equivalence Class Testing

- a) Bagi domain Input ke dalam beberapa kelas yang nantinya akan dijadikan sebagai kasus uji
- b) kelas yang telah terbentuk disajikan sebagai kondisi input dalam kasus uji
- c) Kelas tersebut merupakan himpunan nilai-nilai yang valid dan tidak valid
- d) kondisi input bisa merupakan suatu range, harga khusus, suatu himpunan, atau suatu boolean
- e) Bila kondisi input berupa suatu range, maka input kasus ujinya satu valid dan dua yang invalid
- f) Bila kondisi input berupa suatu harga khusus, maka input kasus ujinya satu valid dan dua yang invalid
- g) Bila kondisi input berupa suatu anggota himpunan, maka input kasus ujinya satu valid dan dua yang invalid

- h) Bila kondisi input berupa suatu anggota Boolean , maka input kasus ujinya satu valid dan dua yang invalid

### **Sample Testing**

- a) Melibatkan sejumlah nilai yang dipilih dari data masukan kelas ekivalensi
- b) Integrasikan nilai tersebut ke dalam kasus uji
- c) Nilai yang dipilih dapat berupa konstanta atau variabel Limit Testing
- d) Kasus uji yang memproses nilai batas (atau titik singular)
- e) Nilai batas disimpulkan dari kelas ekivalensi dengan mengambil nilai yang sama atau mendekati nilai yang membatasi kelas ekivalensi tersebut
- f) Limit test also juga melibatkan data keluaran dari ekivalensi kelas
- g) Pada kasus segi tiga, misalnya limit testing mencoba untuk mendeteksi apakah  $a+b \geq c$  dan bukan  $a + b > c$
- h) Bila kondisi input menentukan suatu range, maka kasus ujinya harus mencakup pengujian nilai batas dari range dan nilai invalid yang dekat dengan nilai batas. Misal bila rangenya antara  $[-1.0, +1.0]$ , maka input untuk kasus ujinya adalah  $-1.0, 1.0, -1.001, 1.001$
- i) Bila kondisi inputnya berupa harga khusus kasus ujinya harus mencakup nilai minimum dan maksimum. Misal suatu file dapat terdiri dari 1 to 255 record, maka kasus ujinya harus mencakup untuk nilai 0, 1, 255 and 256, atau uji saat keadaan record kosong dan record penuh.

### **Robustness Testing**

Data dipilih dari luar range yang didefinisikan. Tujuan pengujian ini adalah untuk membuktikan tidak adanya kejadian yang katastrofik yang dihasilkan akibat adanya keabnormalan.

### **Behavior Testing**

Suatu pengujian yang hasilnya hanya dapat dievaluasi per sub program, tidak bisa dilakukan per modul

### **Requirement Testing**

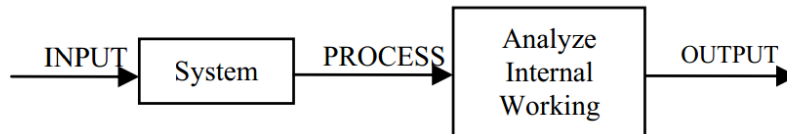
- a) Menyusun kasus uji untuk tiap kebutuhan yang berkorelasi dengan modul / CSU
- b) Tiap kasus uji harus dapat dirunut dengan kebutuhan perangkat lunaknya melalui matriks keterunutan

### **Pengujian White Box**

- a) Metode pengujian dengan menggunakan struktur kontrol program untuk memperoleh kasus uji

b) Dengan menggunakan white box akan didapatkan kasus uji yang :

- Menjamin seluruh jalur independen di dalam modul yang dieksekusi sekurang-kurangnya sekali
- menguji semua keputusan logikal
- menguji seluruh Loop yang sesuai dengan batasannya
- menguji seluruh struktur data internal yang menjamin validitas



Gb.3. Sitem kerja dari Teknik Pengujian White Box

- Basis Path adalah teknik uji coba white box (Tom Mc Cabe).
- Basis Path : untuk mendapatkan kompleksitas logik dari suatu prosedur dan menggunakan ukuran ini sebagai petunjuk untuk mendefinisikan himpunan jalur yang akan diuji
- Basis Path menggunakan notasi graph untuk menggambarkan aliran kontrolnya.

### 1.5 Kesimpulan

Software pengujian adalah teknik penting untuk perbaikan dan pengukuran kualitas sistem perangkat lunak, Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat risiko pada implementasinya.

## Daftar Pustaka

Mohd. Ehmer Khan - *Different Forms of Software Testing Techniques for Finding Error* , tahun 2010

Retno Hendrowati - Perancangan Pengujian Perangkat Lunak Berorientasi Obyek ,2003

Mohd. Ehmer Khan - *Different Approaches to White Box Testing Technique for Finding error* ,tahun 2011

Laurie Williams - *White-Box Testing*, published in 2006

Siegel, Shel., *Object Oriented Software Testing an Hierarchical Approach*, Canada: John Wiley & Sons Inc. 1996