**LAPORAN   
PENENTUAN METODE PENGUJIAN**

**PADA SEBUAH WEBSITE**

****

**Kelompok 4 :**

**Golongan D**

1. Citra Nika Sasmita (E41161716)
2. Al Rizal Fikri S A (E41161930)
3. Ahmad Sofyan F A (E41161935)
4. Ibnu Fadjar (E41161991)

KEMENTRIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
TEKNIK INFORMATIKA  
TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI JEMBER  
2018

METODE PENGUJIAN WEBSITE

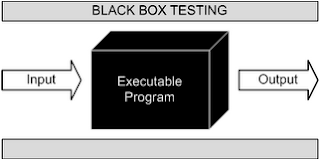
Website : http://www.vipplaza.co.id/

Metode : BLACK BOX

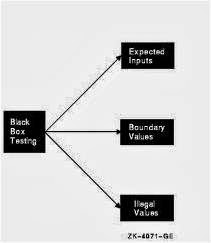
Dasar teori :

Black Box

Pengujian black box merupakan pendekatan komplementer dari teknik white box, karena pengujian black box diharapkan mampu mengungkap kelas kesalahan yang lebih luas dibandingkan teknik white box. Pengujian black box berfokus pada pengujian persyaratan fungsional perangkat lunak, untuk mendapatkan serangkaian kondisi input yang sesuai dengan persyaratan fungsional suatu program.

[](http://4.bp.blogspot.com/-5rF2-HdVIyQ/UpxNfzFu_JI/AAAAAAAAAFc/P9zl3D1-or8/s1600/2.png)

Gambar 1. Black box

[](http://4.bp.blogspot.com/-v7JjFnu_aUI/UpxNkpfr4EI/AAAAAAAAAFs/TJnzRnIPd1w/s1600/3.jpg)

Gambar 2. Yang diuji dalam black box

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitam nya. Sama seperti pengujian black box, mengevaluasi hanya dari tampilan luarnya (interface nya) , fungsionalitasnya, tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui input dan output).

Pengujian black box adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian black box merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.

Pengujian pada Black Box berusaha menemukan kesalahan seperti:

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

Tidak seperti pengujian white box yang dilakukan pada awal proses pengujian, pengujian black box cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian black box memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi. Pengujian black box harus dapat menjawab pertanyaan sebagai berikut :

a) Bagaimana validitas fungsional diuji

b) Kelas input apa yang akan membuat kasus pengujian menjadi lebih baik

c) Apakah system akan sangat sensitive terhadap harga input tertentu

d) Bagaimana batasan dari suatu data diisolasi

e) Kecepatan data apa dan volume data apa yang akan ditoleransi oleh system

f) Apa pengaruh kombinasi tertentu dari data terhadap system operasi.

2.1 Graf Based Testing

Langkah pertama pada pengujian black box adalah memahami objek yang terdapat dalam model perangkat lunak dan menentukan hubungan yang dimiliki antara objek-objek tersebut. Pengujian berbasiskan model graf dilakukan terhadap perilaku system.

Graf Based Testing menggambarkan graf yang mewakili hubungan antar objek pada modul sehingga tiap objek dan hubungannya dapat diuji. Pengujian ini dimulai dari mendefinisikan semua simpul dan bobot simpul, dimana objek dan atribut diidentifikasikan, serta memberikan indikasi titk mulai dan berhenti.

Tabel 1. Notasi Pengujian berbasis graf

|  |  |
| --- | --- |
| Notasi | Arti |
|  | Simpul atau node  Menggambarkan suatu objek |
|  | Link  Menggambarkan hubungan antar objek |
| *X* | Node weight  Menggambarkan properti atau nilai dari data |
| *X* | Link weight  Menggambarkan karakteristik link |
|  | Link paralel  Menggambarkan hubungan yang berbeda yang dibangun antar simpul |
|  | Link simetris  Menggambarkan hubungan antara dua objek |

Terdapat tiga pola link weight, yaitu :

1. Transitivitas, yaitu hubungan antara tiga objek atau lebih yang menentukan bagaimana pengaruh hubungan tersebut menyebar pada objek yang ditentukan
2. Simetris, yaitu hubungan antara dua objek secara dua arah
3. Refleksif, yaitu hubungan yang mengarah pada node itu sendiri atau loop null

Beizer menyebutkan beberapa metode pengujian black box yang menggunakan graf, yaitu :

1. Transaction Flow Modeling, metode ini menggunakan node sebagai representasi langkah pada transaksi, dan link sebagai representasi hubungan logika antara langkah-langkah tersebut
2. Finite state modeling, metode ini menggunakan node sebagai representasi status dan link sebagai representasi transisi. Statechart atau state trasnsition diagram dapat digunakan untuk membuat graf.
3. Data flow modeling, metode ini menguunakan node sebagai represanti objek data dan link sebagai transformasi dari satu objek data ke objek data yang lain.
4. Timing modeling, metode ini menggunakan node sebagai representasi objek program dan link sebagai hubungan sekuensial antara objek.

2.2 Equivalence Partitioning (Partisi ekuivalensi)

Partisi ekuivalensi adalah metode yang membagi domain input dari suatu program ke dalam kelas data, menentukan kasus pengujian dengan mengungkapkan kelas-kelas kesalahan, sehingga akan mengurangi jumlah keseluruhan kasus pengujian.

Bila suatu link weight mempunyai pola transitivitas, simetris, dan refleksif maka akan terdapat kelas ekuivalensi. Kelas ekuivalensi merepresentasikan serangkaian kondisi valid dan invalid untuk kondisi inputan. Secara khusus, suatu kondisi input dapat berupa harga numeric, suatu rentang harga, serangkaian harga yang terkait, atau suatu kondisi Boolean.

Penentuan Kelas Ekuivalensi

1. Bila kondisi input menentukan suatu range, maka satu kelas ekuvalensi valid dan dua yang invalid ditentukan
2. Bila suatu kondisi input memerlukan suatu harga khusus, maka satu kelas ekuivalensi valid dan dua yang invalid ditentukan
3. Bila suatu kondisi menentukan anggota suatu himpunan, maka satu kelas ekuivalensi valid atau dua yang invalid ditentukan
4. Bila suatu kondisi input adalah Boolean, maka satu kelas valid dan satu yang lain ditentukan.

Contoh :

Nomor telepon untuk SLJJ

Ketentuan :

Kode area : kosong atau ada 3 – 4 digit yang dimulai dengan 0

Prefiks : 3 digit dengan tidak dimulai dengan 1 atau 0

Sufiks : 4 digit

Kondisi input :

Kode area : kondisi input Boolean – boleh ada boleh tidak ada

kondisi input range – nilai >200 atau nilai < 999

Prefiks : kondisi input range – nilai > 200 atau nilai < 999

Sufiks : kondisi input harga – 4 digit

2.3 Boundary Value Analysis (Analisis Nilai Batas)

Analisis nilai batas adalah teknik desain proses yang melengkapi partisi ekuivalensi, dengan berfokus pada domain output.

Pedoman untuk menentukan analisis nilai batas :

1. Bila suatu kondisi input mengkhususkan suatu range dibatasi oleh nilai a dan b, maka pengujian harus didesain dengan nilai a dan b, persis di atas dan di bawah a dan b secara bersesuaian
2. Bila suatu kondisi input mengkhususkan sejumlah nilai, maka pengujian harus dikembangkan dengan menggunakan jumlah minimum dan maksimum. Nilai tepat di atas dan di bawah minimum dan maksimum juga diuji.
3. Pedoman 1 dan 2 juga diaplikasikan ke kondisi output.
4. Bila struktur data program telah memesan suatu batasan, maka pengujian akan dilakukan sesuai dengan batasan struktur data terebut.

2.4 Comparison Testing

Pengujian perbandingan adalah metode pembangkitan data uji yang dilakukan pada perangkat lunak yang dibuat redundan. Perangkat lunak yang redundan mempunyai dua tim pengembang yang masing-masing mengembangkan perangkat lunak sendiri-sendiri untuk spesifikasi yang sama.

Metode pengujian perbandingan digunakan untuk membangkitkan data uji untuk salah satu perangkat lunak, yang kemudian digunakan sebagai masukan pada pengujian perangkat lunak yang lain.

Comparison Testing digunakan untuk system yang menganut redundancy, kasus uji yang dirancang untuk satu versi perangkat lunak dijadikan masukan pada pengujian versi perangkat lunak lainnya, dan diharapkan hasil kedua versi perangkat lunak harus sama. Jika hasil pengujian kedua perangkat lunak tersebut berbeda maka kedua perangkat lunak itu akan dicek untuk mencari yang salah.

Dokumentasi komponen software, mencangkup pemeriksaan dokumen dari software itu sendiri, yaitu :

1. Flowchart yang dibuat
2. Deskripsi input yang digunakan
3. Deskripsi output yang digunakan
4. Deskripsi output yang dihasilkan
5. Kesesuaian penulisan (akurasi)
6. Kontrol/kendali terhadap sistem yang dibuat

Strategi Black Box System, meliputi :  
**Batasan nilai untuk testing**, meliputi beberapa nilai, yaitu:

1. Nilai minimum variabel input
2. Nilai di atas nilai minimum
3. Nilai normal
4. Nilai di bawah nilai maksimum
5. Nilai maksimum

**Equivalent Class Testing,**yaitu mengelompokkan input yang direpresentasikan sebagai hasil yang valid atau invalid. Contoh :  
Rekruitasi pegawai berdasarkan pengalaman kerja :  
<1thn    : diterima, part time  
1-3 thn  : diterima, sebagai tenaga kerja profesional  
>4 thn  : diterima, sebagai pegawai tetap

Kesalahan yang dapat terdeteksi melalui testing ini ialah :

1. kebenaran dokumentasi
2. akses basis data
3. hasil akhir program

Kelebihan black box testing :

1. Spesifikasi program dapat ditentukan di awal
2. Dapat digunakan untuk menilai konsistensi program
3. Testing dilakukan berdasarkan spesifikasi
4. Tidak perlu melihat kode program secara detail

Kekurangan black box testing :

1. Bila spesifikasi program yang dibuat kurang jelas dan ringkas, maka akan sulit membuat dokumentasi setepat mungkin

Mengapa menggunakan metode black box?

1. Pengujian yang akan dilakukan tidak pada proses pengembangan, tetapi sudah masuk ke tahap pemakaian. Sehingga, melakukan pengujian berdasarkan apa yang dilihat. Hanya fokus pada fungsionalitas dan output.
2. Pengujian lebih ditujukan pada desain software sesuai standar dan reaksi apabila terdapat kesalahan pada program aplikasi.
3. Penguji tidak harus tahu tentang logic program yang digunakan untuk membangun website tersebut, karena hanya berfokus padafungsionalitas dan output.