



Rekayasa Perangkat Lunak

Pertemuan 9

Teknik Pengujian Perangkat Lunak





Definisi

- Pengujian adalah proses untuk menemukan error pada perangkat lunak sebelum di-delivery kepada pengguna.
- Pengujian P/L menghabiskan upaya 30-40% dari total pekerjaan proyek.
- Untuk proyek yang membahayakan nyawa manusia, biaya pengujian bisa 3-5 X proyek biasa.



Tujuan Pengujian

- Menjalankan program untuk menemukan error.
- Test case yang bagus adalah yang memiliki kemungkinan terbesar untuk menemukan error yang tersembunyi.
- Pengujian yang sukses adalah yang berhasil menemukan error yang tersembunyi.





Prinsip Pengujian

- Harus bisa dilacak hingga sampai ke kebutuhan customer.
- Harus direncanakan sejak model dibuat.
- Menggunakan prinsip Pareto
- Dari lingkup kecil menuju yang besar.
- Tidak bisa semua kemungkinan diuji.
- Dilakukan oleh pihak ketiga yang independen.





Yang berhubungan dengan pengujian

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





Kualitas Pengujian yang baik

- Mencakup semua kemungkinan skenario pengoperasian perangkat lunak
- Mencakup sebanyak mungkin jalur yang dibentuk dari struktur program
- Tidak terlalu sederhana dan tidak terlalu rumit





Karakteristik perangkat lunak yang diuji :

- OPERABILITAS, semakin baik dia bekerja semakin efisien dia dapat diuji.
- OBSERVABILITAS, apa yang anda lihat adalah apa yang anda uji.
- KONTROLABILITAS, semakin baik kita dapat mengontrol perangkat lunak semakin banyak pengujian yang dapat diotomatisasi dan dioptimalkan.
- DEKOMPOSABILITAS, dengan mengontrol ruang lingkup pengujian kita dapat lebih cepat mengisolasi masalah dan melakukan pengujian kembali.





- KESEDERHANAAN, semakin sedikit yang diuji semakin cepat pengujian.
- STABILITAS, semakin sedikit perubahan semakin sedikit gangguan pengujian.
- KEMAMPUAN DIPAHAMI, semakin banyak informasi yang dimiliki semakin detail pengujiannya.



Test Case

- **Test case 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
- **Metode Test Case:** Metode yang dapat membantu memastikan kelengkapan pengujian dan memberikan kemungkinan tertinggi untuk mengungkapkan kesalahan pada perangkat lunak



White box testing / glass box testing

- Adalah pengujian untuk memperlihatkan cara kerja dari produk secara rinci sesuai dengan spesifikasinya
- Metode pengujian dengan menggunakan struktur kontrol program untuk memperoleh kasus uji





Akan didapatkan kasus uji

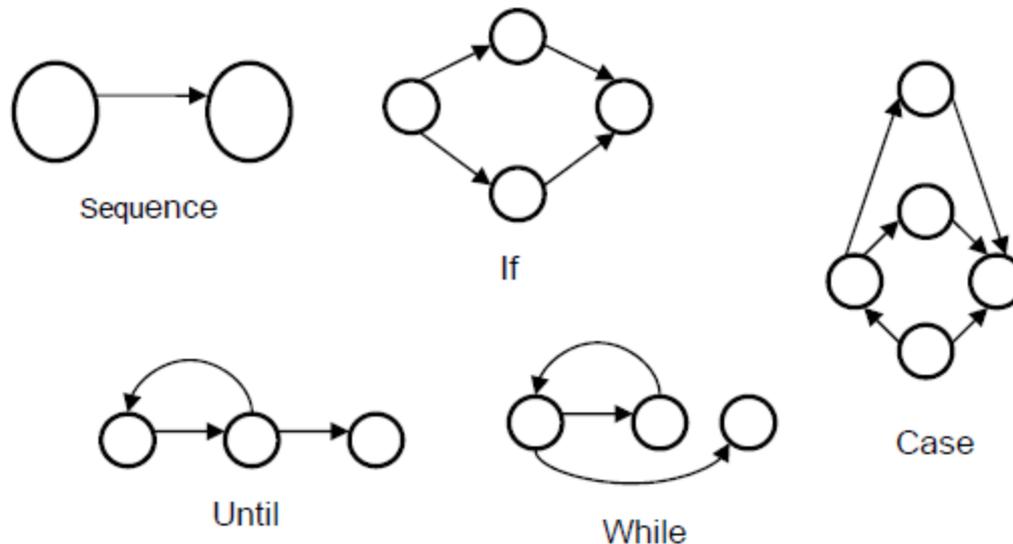
- Menjamin seluruh jalur independen di dalam modul yang dieksekusi sekurang-kurangnya sekali
- Menguji semua keputusan logikal
- Menguji seluruh Loop yang sesuai dengan batasannya
- Menguji seluruh struktur data internal yang menjamin validitas

Secara sekilas dapat diambil kesimpulan white box testing merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

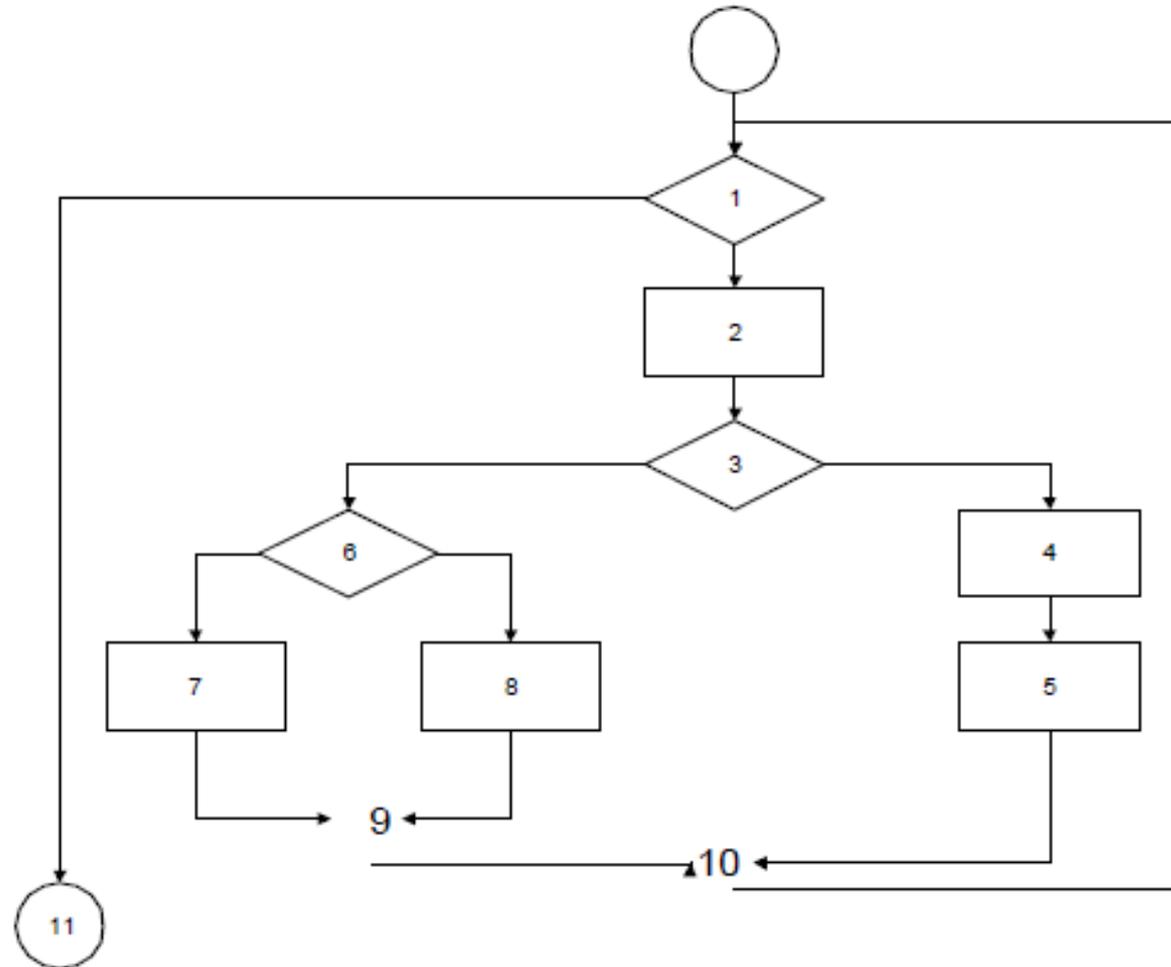


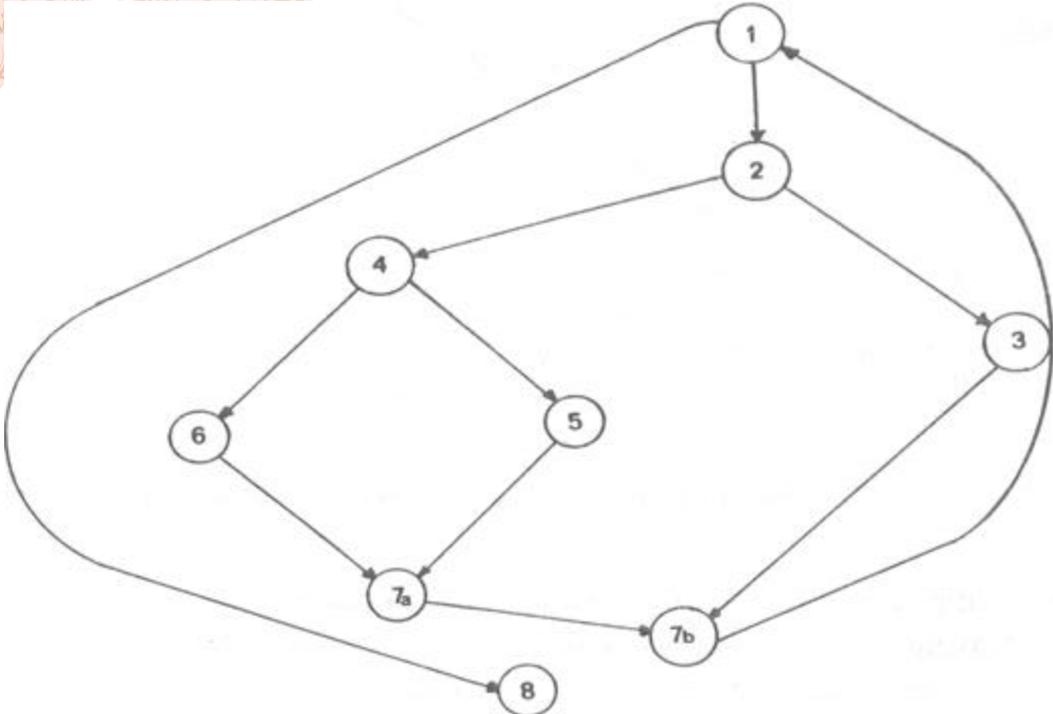
Basis Path Testing

- Untuk mendapatkan kompleksitas yang logis dari suatu prosedur dan untuk mendefinisikan himpunan jalur yang akan diuji
- Basis Path menggunakan notasi graph untuk menggambarkan aliran kontrolnya.



Contoh perancangan prosedural dalam bentuk flowchart





```

1: do while record
    Masih ada baca record
2: if record ke 1 = 0
3: then proses record
    simpan di buffer
    naikan kounter
4: else if record ke 2=0
5: then reser kounter
6: proses record
    simpan pada file
7a: endif
    endif
7b: enddo
8 : end
  
```





Lingkaran/node :

- menggambarkan satu/lebih perintah prosedural.
- Urutan proses dan keputusan dapat dipetakan dalam satu node.

Tanda panah/edge :

- menggambarkan aliran kontrol. Setiap node harus mempunyai tujuan node

Region :

- adalah daerah yg dibatasi oleh edge dan node. Termasuk daerah diluar grafik alir.





BLACK-BOX Testing

- Pengujian ini memungkinkan analisis sistem memperoleh kumpulan kondisi input yang akan mengerjakan seluruh keperluan fungsional program.

Tujuan metode ini mencari kesalahan pada:

- Fungsi yang salah atau hilang
- Kesalahan pada interface
- Kesalahan pada struktur data atau akses database
- Kesalahan performansi
- Kesalahan inisialisasi dan tujuan akhir





Tipe dari Black Box Testing :

- Equivalence class partitioning
- Sample testing
- Limit testing
- Robustness testing
- Behavior testing
- Requirement testing





Tugas

Carilah penjelasan tentang tipe Black-Box dan white box dibawah ini !!

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

