PERTEMUAN KE-14
PENGUJIAN UNIT, VALIDASI, INTEGRASI DAN SISTEM

14.1 TUJUAN PEMBELAJARAN

Adapun tujuan pembelajaran yang akan dicapai sebagai berikut:

14.1. Uji coba unit, uji coba validasi, uji coba integrasi dan uji coba sistem

14.2 URAIAN MATERI

Tujuan Pembelajaran 14.1:
Uji coba unit, uji coba validasi, uji coba integrasi dan uji coba sistem

1. PENGUJIAN UNIT

Unit testing (uji coba unit) fokusnya pada usaha verifikasi pada unit terkecil dari desain PL, yakni modul. Uji coba unit selalu berorientasi pada white box testing dan dapat dikerjakan paralel ayau beruntun dengan modul lainnya.

A. Pertimbangan Pengujian Unit

Interface diuji cobakan untuk menjamin informasi yang masuk atau yang ke luar dari unit program telah tepat atau sesuai denganyang diharapkan. Yang pertama diuji coba adalah interface karena diperlukan untuk jalannya informasi atau data antar modul.

Myers mengusulkan checklist untuk pengujian interface:

1. Apakahjumlah parameter input sama dengan jumlah argumen?
2. Apakah antara atribut dan parameter argumen sudah cocok?
3. Apakah antara sistem satuan parameter dan argumen sudah cocok?
4. Apakah jumlah argumen yang ditransmisikan ke modul yang dipanggil sama dengan jumlah parameter?
5. Apakah atribut dari argumen yang ditransmisikan ke modul yang dipanggil sama dengan atribut parameter?

6. Apakah sistem unit dari argumen yang ditransmisikan ke modul yang dipanggil sama dengan sistem satuan parameter?
7. Apakah jumlah atribut dari urutan argumen ke fungsi-fungsi built-in sudah benar?
8. Adakah referensi ke parameter yang tidak sesuai dengan pain entri yang ada?
9. Apakah argumen input-only diubah?
10. Apakah definisi variabel global konsisten dengan modul?
11. Apakah batasan yang dilalui merupakan argumen?

Bila sebuah modul melakukan I/O ekstemal, maka pengujian interface tambahan harus dilakukan dengan menjawab pertanyaan:

a. Atribut file sudah benar?
b. Pemyataan OPEN/CLOSE sudah benar?
c. Spesifikasi format sudah cocok dengan pernyataan I/O?
d. Ukuran buffer sudah cocok dengan ukuran rekaman?
e. File dibuka sebelum penggunaan?
f. Apakah kondisi End-of-File ditangani?
g. Kesalahan I/O ditangani?
h. Adakah kesalahan tekstual di dalam informasi output?

Kesalahan yang umum di dalam komputasi adalah:

a. kesalah-pahaman atau prosedur aritmatik yang tidak benar
b. operasi mode yang tercampur
c. inisialisasi yang tidak benar
d. inakurasi ketelitian
e. representasi simbolis yang tidak benar dari sebuah persamaan.

Test case harus mengungkap kesalahan seperti

1. perbandingan tipe data yang berbeda
2. preseden atau operator logika yang tidak benar

3. pengharapan akan persamaan bila precision error membuat persamaan yang tidak mungkin
4. perbandingan atau variabel yang tidak benar
5. penghentian loop yang tidak ada atau tidak teratur
6. kegagalan untuk keluar pada saat terjadi iterasi divergen
7. variabel loop yang dimodifikasi secara tidak teratur.

B. Prosedur Pengujian Unit

Program sumber telah dikembangkan, ditunjang kembali dan diverifikasi untuk sintaksnya, maka perancangan test case dimulai. Peninjauan kembali perancangan informasi akan menyediakan petunjuk untuk menentukan test case. Karena modul bukan program yang berdiri sendiri maka driver (pengendali) dan atau stub PL harus dikembangkan untuk pengujian unit.Driver adalah program yang menerima data untuk test case dan menyalurkan ke modul yang diuji dan mencetak hasilnya.Stub melayani pemindahan modul yang akan dipanggil untuk diuji.

2. UJI COBA VALIDASI

Setelah semua kesalahan diperbaiki maka langkah selanjutnya adalah validasi terting. Pengujian validasi dikatakan berhasil bila fungsi yang ada pada PL sesuai denganyang diharapkan pemakai.Validasi PL merupakan kumpulan seri uji coba black box yang menunjukkan sesuai denganyang diperlukan.

Kemungkinan kondisi setelah pengujian:

1. Karakteristik performansi fungsi sesuai dengan spesifikasi dan dapat diterima.
2. Penyimpangan dari spesifikasi ditemukan dan dibuatkan daftar penyimpangan.

Apabila PL dibuat untuk pelanggan maka dapat dilakukan aceeptance test sehingga memungkinkan pelanggan untuk memvalidasi seluruh keperluan. Test ini dilakukan karena memungkinkan pelanggan menemukan kesalahan yang lebih rinci dan membiasakan pelanggan memahami PL yang telah dibuat.

Sebelum perangkat lunak dirilis, dilakukan pengujian alpha dan beta, yaitu:

1. Pengujian Alpha

   Dilakukan pada sisi pengembang oleh seorang pelanggan. PL digunakan pada setting yang natural dengan pengembang "yang memandang" melalui bahu pemakai dan merekam semua kesalahan dan masalah pemakaian.

2. Pengujian Beta

   Dilakukan pada satu atau lebih pelanggan oleh pemakai akhir PL dalam lingkungan yang sebenarnya, pengembang biasanya tidak ada pada pengujian ini. Pelanggan merekan semua masalah (real atau imajiner) yang ditemui selama pengujian dan melaporkan pada pengembang pada interval waktu tertentu.

3. PENGUJIAN INTEGRASI

Pengujian terintegrasi adl teknik yang sistematis untuk penyusunan struktur program, pada saat bersamaan dikerjakan uji coba untuk memeriksa kesalahan yang nantinya digabungkan dengan interface.

Metode pengujian integrasi yaitu:

- top down integration
- buttom up integration

A. TOP DOWN INTEGRATION

Merupakan pendekatan inkrmental untuk penyusunan struktur program. Modul dipadukan dengan bergerak ke bawah melalui kontrol hirarki dimulai dari modul utama.Modul subordinat ke modul kontrol utama digabungkan ke dalam struktur baik menurut depth first atau breadth first.

Proses integrasi:

1. modul utama digunakan sebagai test driver dan stub yang menggantikan seluruh modul yang secara langsung berada di bawah modul kontrol utama.
2. Tergantung pada pendekatan perpaduan yang dipilih (depth / breadth)

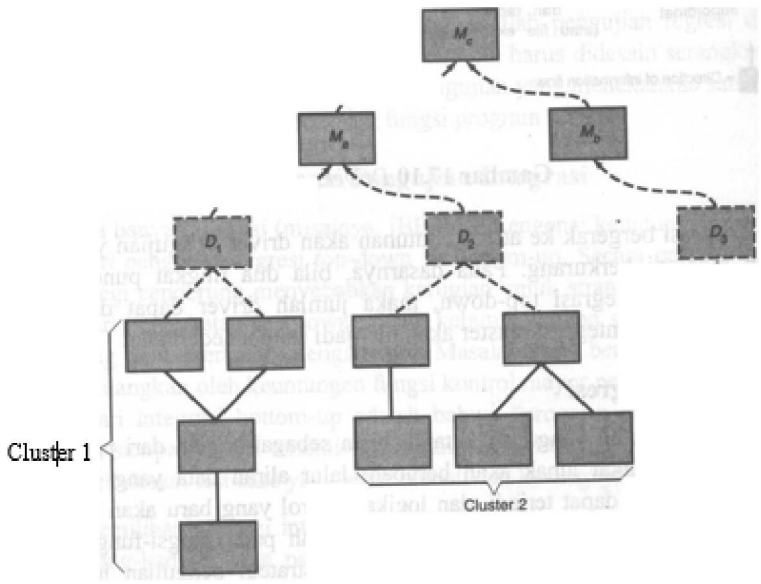
3. Uji coba dilakukan selama masing-masing modul dipadukan
4. Pada penyelesaian masing-masing uji coba stub yang lain dipindahkan dengan modul sebenarnya.
5. Uji coba regression yaitu pengulangan pengujian untuk mencari kesalahan lain yang mungkin muncul.

B. BOTTOM UP INTEGRATION

Pengujian buttom up dinyatakan dengan penyusunan yang dimulai dan diujicobakan dengan atomic modul (yi modul tingkat paling bawah pd struktur program). Karena modul dipadukan dari bawah ke atas, proses yang diperlukan untuk modul subordinat yang selalu diberikan harus ada dan diperlukan untuk stub yang akan dihilangkan.

Strategi pengujian :

1. Modul tingkat bawah digabungkan ke dalam cluster yang memperlihatkan subfungsi PL
2. Driver (program kontrol pengujian) ditulis untuk mengatur input test case dan output
3. Cluster diuji
4. Driver diganti dan cluster yang dikombinasikan dipindahkan ke atas pada struktur program



Gambar 14. 1Buttom Up Integration

5. UJI COBA SISTEM

Pada akhirnya PL digabungkan dengan elemen sistem lainnya dan rentetan perpaduan sistem dan validasi tes dilakukan. Jika uji coba gagal atau di luar skope dari proses daur siklus pengembangan sistem, langkah yang diambil selama perancangan dan pengujian dapat diperbaiki. Keberhasilan perpaduan PL dan sistemyang besar merupakan kuncinya.

Sistem testing merupakan rentetan pengujian yang berbeda-beda dengan tujuan utama mengerjakan keseluruhan elemen sistemyang dikembangkan.

Pengujian sistem antara lain:

1. Recovery Testing adalah sistem testing yang memaksa PL mengalami kegagalan dalam bermacam-macam cara dan memeriksa apakah perbaikan dilakukan dengan tepat.
2. Security Testing adalah pengujian yang akan melalukan verifikasi dari mekanisme perlindungan yang akan dibuat oleh sistem, melindungi dari hal-hal yang mungkin terjadi.
3. Strees Testing. Dirancang untuk menghadapi situasi yang tidak normal pada saat program diuji. Testing ini dilakukan oleh sistem untuk kondisi seperti volume data yang tidak normal (melebihi atau kurang dari batasan) atau frekuensi.

14.3 LATIHAN SOAL/TUGAS

1. Buatlah contoh pengujian unit, dan pengujian integrasi!

14.4 DAFTAR PUSTAKA

1. Pressman, R. S. (2015). Software Engineering. A Practitioner's Approach (8th ed.). New York: McGraw-Hill Education.
2. Sommerville, I. (2011). Software Engineering (9th ed.). Boston: Addison-Wesley.