

PERTEMUAN KE-15

PENGUJIAN BERORIENTASI OBJEK

15.1 TUJUAN PEMBELAJARAN :

Adapun tujuan pembelajaran yang akan dicapai sebagai berikut:

- 15.1. Uji coba berorientasi objek
- 15.2. Pengamatan (Inspections)

15.2 URAIAN MATERI

Tujuan Pembelajaran 15.1:

Uji coba berorientasi objek

Proses ujicoba sistem yang berorientasi objek (object-oriented system) dimulai dengan meninjau ulang analisis dan model desain berorientasi obyeknya (object-oriented analysis and design models). Ketika sebuah program telah dituliskan, object-oriented testing (OOT) dimulai dengan menguji "in the small" dengan class testing (class operations dan collaborations).

Ketika class-class tersebut diintegrasikan menjadi sebuah subsistem, maka masalah kolaborasi class akan diketahui. Terakhir, use-cases dari model OOA digunakan untuk menemukan kesalahan validasi software.

OOT hampir mirip dengan ujicoba software konvensional dalam hal kasus uji yang akan dibangun untuk melatih class-class yang ada dan kolaborasi antar class-nya juga perilakunya. OOT berbeda dari ujicoba software konvensional dalam hal penekanan terhadap konsistensi dan kelengkapan penaksiran dari model OOA dan OOD yang telah dibangun. OOT cenderung lebih fokus kepada masalah integrasi dari pada unit testing.

Aktivitas pengujian berorientasi objek:

1. Meninjau ulang model OOA dan OOD
2. Ujicoba Class setelah penulisan program sumber
3. Ujicoba Integrasi dalam subsistems
4. Ujicoba Integrasi subsistem yang telah ditambahkan kedalam sistem

5. Ujicoba validasi berdasarkan OOA use-cases

Dalam pengujian berorientasi objek, unit terkecil adalah objek dan class, sistem merupakan sekumpulan komunikasi-komunikasi antar objek. Class sering dirancang untuk menerima urutan pesan-pesan tertentu yang mengakibatkan respon class terhadap pesan-pesan tersebut menjadi berbeda-beda. Hal ini selanjutnya disebut dengan perilaku (behavior) class. Perilaku tersebut dapat dikendalikan dengan nilai yang ter-enkapsulasi, urutan pesan atau keduanya.

Pengujian adalah suatu proses pengeksekusian program yang bertujuan untuk menemukan kesalahan. Pengujian sebaiknya menemukan kesalahan yang tidak disengaja dan pengujian dinyatakan sukses jika berhasil memperbaiki kesalahan tersebut. Selain itu, pengujian juga bertujuan untuk menunjukkan kesesuaian fungsi-fungsi perangkat lunak dengan spesifikasinya.

Pengujian dapat dikategorikan atas :

1. Pengujian terhadap proses pengembangan sistem dan dokumendokumen pendukung. Proses berarti sejumlah aktivitas yang didukung oleh dokumen yang mendeskripsikan aktivitas-aktivitas.
2. Pengujian terhadap analisis dan model perancangan. Dalam sistem berorientasi objek, pengujian model analisis dan perancangan adalah hal yang sangat penting.
3. Pengujian secara statik dan dinamik untuk implementasi. Tujuannya adalah mencari kesalahan sedini mungkin dalam proses, tetapi kesalahan dalam kode untuk sistem yang besar dan kompleks tidak dapat dihindarkan. Pengujian statik merupakan inspeksi kode untuk menemukan kesalahan logic. Pengujian dinamik merupakan eksekusi dengan data uji untuk menemukan kesalahan dalam kode.

Perangkat lunak berorientasi objek berbeda dari perangkat lunak procedural (konvensional) dalam hal analisis, perancangan, struktur dan teknik-teknik pengembangannya. Bahasa pemrograman berorientasi objek mempunyai ciri-ciri adanya pembungkusan (encapsulation), keanekaragaman (polymorphism), dan pewarisan (inheritance) yang membutuhkan dukungan pengujian tertentu. Fokus

pengujian perangkat lunak berorientasi objek dimulai pada hasil analisisnya, dilanjutkan pada hasil perancangannya, dan diakhir pada hasil pemrogramannya. Model yang dihasilkan pada analisis dan perancangan harus diperiksa terutama dalam hal:

1. Semantic correctness, yaitu kesesuaian model dengan domain permasalahan di dunia nyata. Jika model merefleksikan dunia nyata secara akurat, berarti model tersebut benar secara semantik, dan
2. Consistency, yaitu kesesuaian kelas dengan objek turunannya maupun kesesuaian asosiasi kelas dengan kelas lainnya.

Tujuan Pembelajaran 15.2:

Pengamatan (Inspections)

Inspection, inspeksi meliputi aktifitas seperti pengukuran, pemeriksaan, dan verifikasi untuk menentukan apakah pekerjaan dan sasaran mencapai persyaratan dan kriteria penerimaan produk. Inspeksi disebut juga review, review produk, audit.

Inspeksi merupakan suatu kegiatan penilaian terhadap suatu produk berdasarkan metode dan standar yang sudah ditentukan. Penilaian bisa berupa produk tersebut baik atau rusak, dapat juga berupa penentuan tahapan pengembangan atau pembangunan perangkat lunak dapat diterima atau ditolak. Dapat diperjelas bahwa inspeksi perangkat lunak merupakan suatu aktifitas operasional untuk memeriksa materi atau spesifikasi perangkat lunak yang diperlukan untuk dapat memenuhi materi atau spesifikasi pada proses tahapan pengembangan atau pembangunan perangkat lunak berikutnya atau pemenuhan materi atau spesifikasi yang dibutuhkan sebelum dipergunakan. Inspeksi perangkat lunak Inspeksi berguna untuk untuk quality control / pengendalian mutu dan jaminan kualitas. Hal ini sesuai fungsinya untuk memeriksa tiap-tiap materi atau spesifikasi yang dihasilkan dengan membandingkan dengan standarnya / kebutuhannya, sehingga akan terdeteksi ketidaksesuaian terhadap hasil yang diperoleh. Inspeksi perangkat lunak dapat berupa pemeriksaan dan pengujian pada tahapan pengembangan atau pembangunan perangkat lunak agar tidak terjadi

pengulangan tahapan yang berdampak pada waktu penyelesaian pengembangan atau pembangunan perangkat lunak. Pemeriksaan dan pengujian ini dilakukan terhadap ketidak konsistenan antara program dan spesifikasi, dan dilakukan terhadap kinerja dan kehandalan program pada kondisi operasional.

Inspeksi perangkat lunak akan lebih efektif bila, menganalisis dan memeriksa representasi sistem seperti dokumen persyaratan, diagram rancangan, dan kode sumber program. Inspeksi ini dapat diterapkan pada semua tahap proses dalam menemukan kesalahan program. Construction aktivitas yang mengkombinasikan pembangkitan kode dan pengujian yang diperlukan untuk menemukan kesalahan dalam kode program. Integration proses integrasi antara modul-modul program sehingga menjadi suatu sistem yang lengkap. Testing diperlukan untuk menemukan kesalahan-kesalahan yang ada. Inspeksi ini dapat dilengkapi dengan beberapa analisis otomatis teks sumber sistem atau dokumen terkait. Pemeriksaan perangkat lunak dan analisis terotomasi merupakan teknik verifikasi dan validasi statis karena tidak menuntut sistem dieksekusi. Pada tahap ini terdapat dua tipe pengujian yang dapat digunakan pada berbagai tahap proses perangkat lunak yaitu:

1. Pengujian cacat yang ditujukan untuk menemukan ketidak konsistenan antara program dan spesifikasinya. Ketidakkonsistenan ini pada umumnya dikarenakan kesalahan atau cacat program.
2. Pengujian statistik, dipakai untuk menguji kinerja dan keandalan program dan memeriksa bagaimana kerjanya pada kondisi operasional.

Kesalahan (error) yang tidak terdeteksi oleh inspeksi adalah Logical Error, yaitu kesalahan logika dari program itu sendiri. Ini adalah kesalahan yang paling sulit ditelusuri. Hal ini sering terjadi, walaupun logical error pada program namun masih tetap dapat menghasilkan output yang benar, apabila jika dikompilasi dengan hasil yang diharapkan ternyata tidak sesuai oleh user, dan baru terdeteksi/terlihat pada saat digunakan. Disini perlunya system analyst yang bertugas sebagai menterjemahkan user requirements menjadi system/software specification serta Quality Assurance Specialist yang berperan melakukan validasi dan verifikasi proses dan produk akhir software.

15.3 LATIHAN SOAL/TUGAS

1. Buatlah contoh pengujian pada perangkat lunak berorientasi objek!

15.4 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.