

BAB III

PENGUJIAN SPESIFIK

3.1 Pendahuluan

Saat perangkat lunak komputer menjadi semakin kompleks, kebutuhan akan pendekatan pengujian khusus juga makin berkembang.

Metode pengujian white-box dan black-box dapat diaplikasikan pada semua lingkungan, arsitektur, dan aplikasi, tetapi kadang-kadang dalam pengujian diperlukan pedoman dan pendekatan yang unik.

3.2 Beberapa Pengujian Spesifik

Beberapa pengujian yang terkait di antaranya:

- ✓ Pengujian GUI (Graphical User Interfaces)
- ✓ Pengujian Arsitektur Client / Server
- ✓ Pengujian Dokumentasi dan Fasilitas Help
- ✓ Pengujian Sistem Real-Time

3.2.1 Pengujian GUI (Graphical User Interfaces)

Karena GUI modern memiliki bentuk dan cita rasa yang sama, maka dapat dilakukan sederetan pengujian standar.

Untuk window:

- ☐ Apakah window yang aktif disorot secara tepat?
- ☐ Apakah window akan menutup secara tepat?

Untuk menu pull-down dan operasi mouse:

- ☐ Apakah operasi menu pull-down bekerja secara tepat?
- ☐ Apakah operasi mouse dikenali dengan baik pada seluruh konteks interaktif?

Entry data:

- ☐ Apakah mode grafik dari entry data bekerja dengan baik?
- ☐ Apakah data invalid dikenali dengan baik?
- ☐ Apakah pesan input data sangat pintar?

3.2.2 Pengujian Arsitektur Client / Server

Beberapa hal yang menjadi bahan pertimbangan pengujian ini di antaranya:

- ❑ Sifat terdistribusi dari lingkungan client / server

Sifat terdistribusi (client/server, peer-to-peer, multiple server, proxy server, dsb) berbeda-beda tiap arsitektur.

- ❑ Masalah kinerja yang berhubungan dgn pemrosesan transaksi

Karena pembagian fungsional berbeda, maka kinerjanya juga berbeda-beda.

- ❑ Kehadiran potensial dari sejumlah platform perangkat keras yang berbeda

Efek yang terjadi jika terjadi perubahan penambahan atau pengurangan infrastruktur.

- ❑ Kompleksitas komunikasi jaringan

Jaringan yang lebih kompleks tentu pengujiannya lebih kompleks pula.

- ❑ Kebutuhan akan layanan client multiple dari suatu database terpusat

Kinerja server saat menghadapi banyak permintaan.

- ❑ Persyaratan koordinasi yang dibebankan pada server

Pengaturan server juga perlu diuji untuk kinerja pelayanan yang bebas masalah.

3.2.3 Pengujian Dokumentasi dan Fasilitas Help

Dokumentasi merupakan elemen ketiga dari konfigurasi perangkat lunak. Sering terjadi pengguna mengikuti tuntunan dokumentasi tetapi hasilnya tidak sesuai dengan yang diprediksi oleh dokumen.

Karenanya pengujian dokumentasi harus menjadi suatu bagian yang berarti dari setiap rencana pengujian perangkat lunak.

Pengujian ini dapat didekati dalam dua fase:

- ❑ Kajian teknis formal, yang menguji kejelasan editorial dokumen

- ❑ Live test, menggunakan dokumentasi dalam kaitannya dengan penggunaan program aktual. Pengujiannya dapat menggunakan teknik yang analog dengan berbagai metode pengujian *black-box*.

3.2.4 Pengujian Sistem Real-Time

Banyak aplikasi real-time yang bersifat asinkron dan tergantung waktu.

Pengujian ini mempertimbangkan penanganan kejadian (yakni pemrosesan interupsi), timing data, dan paralelisme tugas-tugas (proses) yang menangani data.

Strategi empat langkah yang dapat digunakan:

- ☐ Pengujian tugas (kinerja tiap fungsi)
- ☐ Pengujian tingkah laku (misalnya tingkah laku antarmuka)
- ☐ Pengujian antar-tugas (hubungan dengan fungsi lainnya)
- ☐ Pengujian sistem (sebagai satu keseluruhan sistem)

3.3 Static and Dinamic Testing

Static testing mengacu pada pengujian saat perangkat lunak tidak sedang dijalankan (*running*). Sedangkan **Dinamic testing** mengacu pada pengujian saat perangkat lunak sedang dijalankan (*running and using the software*).

Jika dianalogikan pengujian kendaraan motor, *Static Testing* misalnya dengan cara memukul-mukul ban, memeriksa cat, dan sebagainya. Sedangkan *Dinamic Testing* misalnya dengan mendengarkan suara mesin, mengendarai tanjakan / turunan, menguji kepakeman rem, dan sebagainya.

Contoh	Static Testing	Dinamic Testing
Black-Box	Menguji spesifikasi dari file yang terkait. Misalnya dengan pengujian apakah ukuran file yang terbentuk jika database mencapai jumlah record tertentu tidak akan melebihi dari yang dipersyaratkan, dsb.	Menguji perilaku program saat dijalankan tanpa mengetahui struktur internalnya. Misalnya pengujian untuk mengetahui perilaku apa yang terjadi jika user salah memasukkan username / password, dsb.
White-Box	Menguji kode program saat program tidak dijalankan. Misalnya dengan melakukan pengujian pada salah satu fungsi di dalam program, dsb.	Menguji program saat program dijalankan dengan lebih teliti. Misalkan pengujian untuk mengetahui apa yang terjadi jika melibatkan data perhitungan yang sangat detail / teliti, dsb. Dapat dilengkapi dg monitor status log secara real time, dsb.

-ooOoo-