

TOOL PENGUJIAN

Dalam duapuluh tahun terakhir sebagian besar usaha pengembangan dikeluarkan dalam proses-proses pengembangan yang mengharuskan adanya aktifitas pengujian terutama secara manual dan seringkali merupakan fungsi yang kurang efisien dalam beberapa organisasi. Walaupun demikian, beberapa metoda pengujian baru sedang dikembangkan untuk meningkatkan efektivitas dan produktivitas pengujian.

Pengujian menghasilkan sejumlah besar informasi, memerlukan sejumlah eksekusi komputer, dan mengharuskan adanya koordinasi dan komunikasi antar sesama pekerja. Tool pengujian dapat memudahkan pelaksanaan uji produksi, uji eksekusi, pengendalian informasi umum, dan komunikasi.

Pemilihan tool yang sesuai merupakan suatu aspek yang sangat penting dalam proses pengujian. Jika teknik yang ada sedikit jumlah dan luas cakupannya maka sebaliknya tool sangat banyak jumlahnya dengan cakupan yang tidak terlalu luas dan masing-masing memiliki kemampuan yang berbeda karena setiap tool dirancang untuk memenuhi suatu objek pengujian yang spesifik..

Pemilihan tool berpengaruh pada efektivitas dan efisiensi pengujian, seperti halnya pemilihan martil yang cocok untuk menancapkan paku dengan ukuran tertentu pada pemilihan bahan tertentu pula. Tingkat efisiensi pengujian yang didapatkan bergantung pada pemilihan tool yang tepat dan sesuai.

OVERVIEW

Tool pengujian adalah alat bantu yang digunakan oleh individu yang bertanggungjawab dalam masalah pengujian. Tool mencakup sejumlah besar aktivitas dan dapat diterapkan dalam semua fase daur hidup pengembangan system. Tool dapat dibedakan menjadi :

- Manual dan Otomatis
- Pengujian Statis dan Pengujian Dinamis
- Evaluasi Struktur Sistem dan Evaluasi Fungsi Sistem

Ada beberapa tingkat keahlian yang berbeda-beda yang diperlukan untuk dapat menggunakan suatu tool. Beberapa tool mengharuskan keahlian yang sangat teknis dan melibatkan pengetahuan secara mendalam tentang komputer dan sistem yang akan diuji. Sebagian lagi sangat umum dan dapat digunakan oleh hampir semua individu yang memiliki tanggung jawab pengujian. Sebagian tool tidak banyak menyita waktu pengujian dan sebagian lagi memerlukan suatu tim yang memanfaatkan perangkat komputer secara intensif untuk dapat menggunakan tool tersebut.

TOOL PENGUJIAN dan PENGGUNAAN TOOL

1. Kriteria Uji Penerimaan :
memberikan standar yang harus dipenuhi sistem agar diterima oleh pengguna
2. Analisis Nilai Batas :
membagi sistem menjadi segmen-segmen logik yang kemudian diuji sesuai batasan tiap-

- tiap segmen (*top-down*)
3. Penggambaran Sebab-Akibat :
membatasi jumlah transaksi uji dengan menentukan jumlah variabel kondisi untuk resiko minimal
 4. *Checklist*:
memberikan sejumlah 'pertanyaan' untuk meraba masalah yang potensial
 5. Perbandingan Kode:
membandingkan 2 versi program yang sama untuk mengidentifikasi perbedaannya
 6. Analisis Berbasis Kompilator :
mendeteksi kesalahan selama proses kompilasi program
 7. Uji Pengukuran Berbasis Kolnpleksitas :
menggunakan metoda matematis dan statistik untuk membentuk persamaan yang dapat mengidentifikasi kompleksitas program
 8. Uji Konfirmasi :
memverifikasi keberadaan suatu kondisi
 9. Analisis Aliran Kendali :
mengidentifikasi inkonsistensi dalam proses (*loop* tanpa henti, percabangan ke tengah rutin, dsb)
 10. Pembuktian Kebenaran:
membentuk sejumlah pernyataan dan hipotesa untuk mengevaluasi kebenaran proses
 11. Uji Pengukuran Berbasis Cakupan :
menggunakan persamaan matematis untuk memperlihatkan derajat kompleksitas proses yang telah diberikan oleh proses uji
 12. Kamus Data:
pembangkitan data uji untuk memveritjkasi program validasi berdasarkan data dalam kamus data yang berisi elemen data dan atributnya
 13. Analisis Aliran Data :
mempertegas pendefinisian data, identifikasi data teridentifikasi tak terpakai atau data tak teridentifikasi tapi terpakai
 14. Uji Fungsional Berbasis Desain :
mengevaluasi fungsi yang perlu untuk mendukung kebutuhan sesuai rancangan
 15. *Review* Desain:
melakukan review pada titik-titik yang telah ditentukan selama pengembangan sistem untuk melihat kemajuan
 16. *Desk Checking*:
evaluasi logik program oleh pemrogram atau analis sistem setelah pengkodean program atau setelah sistem dirancang
 17. UjiBencana :
mensimulasi kegagalan operasi atau sistem untuk menentukan apakah sistem mampu dipulihkan dengan benar setelah kegagalan terjadi dan waktu yang dibutuhkan
 18. Tebak Kesalahan :
membuat transaksi uji yang memiliki kemungkinan kesalahan yang tinggi berdasarkan pengalaman penguji dan data historis
 19. *Executable Specs* :
memberikan interpretasi tingkat tinggi (ditulis dalam bahasa tingkat tinggi) dari spesifikasi sistem untuk membuat hasil / respon dari data uji

20. *Exhaustive Test*:
mencoba membuat transaksi uji untuk tiap kondisi yang mungkin dan setiap jalur program
21. Pencarian Fakta:
mendapatkan fakta-fakta pendukung proses pengujian
22. Diagram Alir :
menggambarkan logika dan aliran data dari sistem
23. Inspeksi:
menjelaskan setiap tahap produk dan diperiksa apakah sesuai dengan kriteria
24. Instrumentasi :
mengukur fungsi struktur sistem dengan pencacahan dan alat pantau lain Fasilitas
25. Fasilitas Uji Terintegrasi :
memungkinkan integrasi data uji dalam lingkungan produksi sehingga pengujian dapat dilakukan selama proses produksi
26. Pemetaan :
mengidentifikasi bagian dari program yang diuji dan frekuensi pengujiannya
27. Pemodelan :
mensimulasikan fungsi lingkungan atau struktur sistem untuk menentukan efisiensi fungsi sistem yang diusulkan
28. Pengoperasian Paralel :
memverifikasi bahwa 2 versi sistem (lama dan baru) memberikan hasil yang sama
29. Simulasi Paralel :
memperkirakan hasil yang diharapkan dari proses dengan mensimulasikan proses
30. *Peer Review* :
menaksir secara langsung efisiensi, gaya, pemenuhan standar dll. dengan tujuan meningkatkan kualitas produk
31. Matriks Resiko :
menghasilkan matriks keterkaitan antar resiko sistem, segmen di mana resiko muncul, dan tersedianya pengendalian terhadap resiko yang muncul
32. SCARF (File Review Audit Kendali Sistem) :
membangun catatan/*history* dari masalah potensial untuk memperbandingkan masalah tersebut untuk jangka waktu tertentu
33. Penilaian :
mengidentifikasi area aplikasi yang memerlukan pengujian dengan mengurutkan kriteria-kriteria yang berkaitan dengan masalah
34. *Snapshot*:
memperlihatkan isi dari *storage* komputer pada titik tertentu selama proses
35. Eksekusi Simbolik :
mengidentifikasi jalur proses dengan menguji program menggunakan data simbolik (bukan data aktual)
36. Catatan Sistem :
memberikan jejak pemeriksaan dari peristiwa yang terpantau
37. Data Uji:
membuat data transaksi yang digunakan menentukan fungsi sistem
38. Pembangkit Data Uji :
memberikan data transaksi uji berdasarkan parameter yang perlu diuji
39. Merunut :

- mengikuti dan mendaftarkan alur pemrosesan dan pencarian basis data
40. Program-program Utilitas : menganalisa dan mencetak hasil pengujian dengan program multi-gunafungsi
 41. Pengujian Volume:
mengidentifikasi batasan sistem dan membuat transaksi bervolume besar yang dirancang untuk melampaui batasan sistem tersebut
 42. *JWalk-Throughs* :
memimpin suatu tim melalui suatu simulasi manual produk menggunakan transaksi uji

Penjelasan lengkap ada dalam Appendix A dan tool yang lebih efektif dibahas lebih rinci pada bab (6-11) mengenai Daur Hidup Pengembangan Sistem (SDLC, System Development Live Cycle).

PEMILIHAN DAN PENGGUNAAN TOOL PENGUJIAN

Kriteria pemilihan tool yang utama menyangkut fungsi spesifik, biaya dan kemampuan pengaplikasiannya. Dewasa ini yang menarik dan banyak dipakai adalah termasuk sistem pendukung uji otomatis (pembangkit data uji) dan peningkatan penggunaan analisis otomatis (analisis berbasis kompilator). Sejalan dengan makin banyaknya teknik formal yang digunakan dalam analisa kebutuhan dan perancangan, dimungkinkan pula peningkatan penggunaan analisa otomatis. Selain itu banyak pula tool analisis canggih yang diaplikasikan pada kode yang sedang dibangun. Makin banyak dilakukan pengendalian menyeluruh dan otomatisasi dari pelaksanaan uji, baik dalam pembangkitan kasus uji maupun dalam pengelolaan proses dan hasil uji

Hal yang penting adalah bahwa validasi selalu muncul dalam seluruh fase daur hidup pengembangan. Satu keunggulan dari teknik manual adalah kemampuan penerapannya yang seragam pada fase analisa kebutuhan, perancangan, dan pengkodean. Tool ini dapat digunakan tanpa biaya besar, namun agar bisa efektif harus dilakukan secara serius dan dengan disiplin tinggi. Beberapa faktor penentu kesuksesan pengujian adalah perencanaan yang hati-hati obyektif uji yang jelas, definisi tool yang tegas, pengelolaan yang baik, penyimpanan catatan yang terorganisir dan komitmen yang kuat. Dalam pemilihan tool yang tepat ada 4 langkah yaitu :

Langkah 1 : Pencocokan Tool sesuai dengan kegunaannya

Tool adalah alat bantu penyelesaian tugas, sehingga semakin baik tool yang dipakai maka semakin efisien tugas diselesaikan- Pemilihan tool yang salah tidak hanya mengurangi efisiensi pengujian tetapi juga dapat menyebabkan obyektif pengujian tidak didapatkan.

Penggunaan dari tiap tool telah diberikan pada tabel tool pengujian dan penggunaan tool Teknik pengujian harus dipilih sebelum pemilihan tool (dalam bab sebelumnya telah dibahas hubungan antara faktor dan teknik pengujian).

Kaitan antara teknik pengujian dan tool yang mendukungnya diperlihatkan pada Fig. 30 Matriks Teknik Pengujian - Tool, hlm. 102-104. Matriks ini dapat digunakan untuk memilih tool yang terbukti paling menguntungkan untuk tiap teknik pengujian, tetapi tidak membatasi pemilihan tool oleh penguji

Langkah 2 : Pemilihan Tool yang sesuai dengan Fase Daur Hidup

Bentuk pengujian bervariasi bergantung fase daur hidup yang diuji dan metoda yang dipakai. Sejalan dengan berkembangnya daur hidup, tool cenderung beralih dari manual ke otomatis. Tetapi hal bukan berarti bahwa tool otomatis selalu lebih efektif daripada tool manual karena kebanyakan pengujian yang produktif terjadi pada fase awal daur hidup dengan menggunakan tool manual.

Keterkaitan antara 6 fase daur hidup dan 42 tool pengujian yang paling cocok dan efektif untuk digunakan dalam fase tersebut diperlihatkan pada Fig. 31 Matriks Fase SDLC - Tool, hlm. 106-112. Matriks ini dapat digunakan dalam tahap kedua untuk memperkecil tool yang teridentifikasi pada langkah pertama dan memberikan tool yang efektif dalam fase dimana pengujian dilakukan.

Langkah 3 : Penyesuaian Tool dengan Tingkat Keahlian Penguji

Penguji harus memilih tool yang sesuai dengan tingkat keahliannya. Bukan berarti bahwa tidak diperlukan pelatihan untuk memakai tool, tetapi mereka harus memiliki keahlian dasar. Penggolongan tool sesuai dengan tingkat keahlian penguji diperlihatkan pada Fig. 32 Tingkat Keahlian untuk Menggunakan Tool Pengujian, hlm. 109-112. Keahlian dibagi dalam 4 tingkat, yaitu :

1. Tingkat Pengguna :
pengetahuan mendalam tentang aplikasi dan segi bisnisnya, termasuk di dalamnya bidang komputerisasi, manajemen, dan pengetahuan untuk mengenali dan menghadapi masalah pengguna.
2. Tingkat Pemrogram :
pemahaman konsep komputer, diagram alir, pemrograman dan bahasa yang dipakai, debugging, dan dokumentasi.
3. Tingkat Sistem:
kemampuan menerjemahkan kebutuhan pengguna ke dalam spesifikasi rancangan sistem komputer, termasuk di dalamnya diagram alir, analisis masalah, metodologi perancangan, operasi komputer, bisnis secara umum, identifikasi dan analisis kesalahan, dan manajemen proyek.
4. Tingkat Teknis :
pemahaman sesuatu yang sangat khusus secara teknis, untuk pengujian mencakup pemrograman sistem, administrasi basis data, statistik, akunting, dan paket perangkat lunak operasi.

Keempat kemampuan di atas dibutuhkan untuk menggunakan tool dan dalam beberapa kasus dibutuhkan tambahan keahlian lain.

Langkah 4 : Pemilihan Tool yang sesuai dengan Anggaran

Pengujian harus dilaksanakan dengan anggaran dan jangka waktu tertentu. Sulit untuk menentukan biaya yang dikeluarkan untuk menggunakan masing-masing tool karena bergantung pada perangkat keras dan lunak yang dipakai. Untuk itu diberikan pengkategorian dari 42 tool ke

dalam 3 tingkat biaya yaitu tinggi, sedang dan rendah seperti diperlihatkan pada Fig. 33 Biaya Penggunaan Tool Pengujian, hlm. 113-114.

Ada kemungkinan setelah 4 langkah dilakukan dan hasilnya tidak ada tool yang terpilih. Dalam hal ini ada 2 pilihan, yaitu :

1. mengulangi proses pemilihan dengan lebih memperhatikan kriteria pemilihan.
2. mengabaikan pemilihan secara formal dan mengandalkan kebijaksanaan serta pengalaman.

Referensi : Perry, W.E; A Structured Approach to Systems Testing; QED Information Science; 1983