PERAN PENTING TESTING DAN QUALITY ASSURANCE DALAM SIKLUS PENGEMBANGAN SISTEM

Hari Mantik
hari.mantik@yahoo.com

Abstrak

Testing merupakan kegiatan pengujian dalam menilai tingkat fungsionalitas dalam suatu aplikasi/program. Mulai dari mana dan sampai sejauh mana seorang tester melakukan pengujian terhadap suatu sistem? Apakah pengujian hanya meneliti tingkat error/bug dari suatu sistem, tanpa melihat dari sudut atau perspektif yang lain?

Posisi pengujian dalam daur hidup pengembangan sistem berada pade fase implementasi, atau fase yang nyaris berada di akhir siklus. Sudah efektif kah itu? Peran seorang QA (quality assurance) yang mencakup quality analyst, quality control, dan tester harus lebih diperluas dan seyogyanya sudah diperkenalkan saat suatu sistem baru dimulai dalam daur hidup sistem (SDLC).

Pendahuluan

Testing bukan hanya sebuah kegiatan pengujian dalam menilai fungsionalitas dalam suatu sistem. *Testing* sebaiknya menjadi suatu alat ukur dalam menentukan kualitas dari sistem, kehandalan dari sistem, dan tentu saja testing harus menciptakan hasil yang memberikan rasa percaya diri bagi seluruh *stakeholders*. Beberapa hal yang membuat testing menjadi keharusan adalah:

1. Menemukan kesalahan/*bug* saat proses pengembangan produk dalam SDLC.
2. Sebagai acuan dalam meningkatkan kepuasan user melalui produk yang handal / reliability.
3. Produk yang berkualitas menciptakan rasa percaya diri / confidence.
4. Menjamin kinerja suatu produk
5. Dan memastikan perusahaan pemilik / *product owner* tetap berjalan serta meningkatkan *business value*.

Definisi dan tinjauan testing

Software testing merupakan suatu metode dalam menilai (*assess*) tingkat fungsionalitas dalam suatu aplikasi perangkat lunak. Testing juga merupakan suatu alat ukur dalam menentukan atau meninjau kualitas, kinerja, ataupun keandalan dari suatu aplikasi perangkat lunak sebelum dilakukan implementasi di ranah publik (*real production*). Pertanyaan yang sering timbul adalah, mengapa testing sangat dibutuhkan? Padahal dalam metodelogi SDLC hanya sedikit yang dibahas mengenai testing.

Pertama, testing dibutuhkan untuk mengukur tingkat *error/bug* selama masa pengembangan (development phase). Tingkat kepercayaan pelanggan sangat tergantung pada kualitas yang dibangun saat masa pengembangan.

Kedua, menghadapi perubahan sistem yang bergerak sangat dinamis dibutuhkan metode pengujian yang berguna sebagai *control* / pengawasan yang paling efektif di dalam masa pengembangan sistem.

Ketiga, kerjasama yang baik antara Project-Developer-Tester-User merupakan kombinasi terbaik dalam meng-hasilkan produk dan sistem yang berkualitas.

Enam hal penting diperlukannya testing dalam pengembangan sistem:

1. Testing memberikan informasi bagi para stakeholders
2. Measure quality, testing menetapkan standar kualitas pada produk,
3. Mitigate risk, testing mengurangi resiko yang di-perkirakan timbul,
4. Testing memberikan rasa percaya diri,
6. Testing memastikan bahwa aplikasi harus sesuai dengan tujuan, dan
7. Testing sebagai satu-satunya alat pencari *error/bug*.

Testing environment / Lingkungan dalam proses pe-ngujian.

Testing sebaiknya dilakukan dalam *multi-environment* / *multi-stage*. Hal ini terkait dengan fungsi-fungsi yang akan di uji, dan peran stakeholder dalam tiap envi-ronment / *stage*. Lingkungan testing terbagi atas tiga *stage*, yaitu:

1. Environment development
2. Environment test
3. Environment production

Environment development, atau yang disebut dengan SIT *stage* (fase *system integration testing*), adalah pengujian yang mencakup *blackbox* testing dan *whitebox* testing. Pengujian dilakukan secara menyeluruh baik oleh team programmer, team developer, dan team testing. Tujuan dari fase ini adalah melihat sistem yang di uji secara dalam, komprehensif dan detail dari sudut pandang teknis (programmer dan developer) dan dari sudut pandang fungsional (tester). Data yang digunakan selama dilakukan pengujian adalah data simulasi.

Environment test, atau yang disebut dengan UAT stage (fase *user acceptance testing*), adalah pengujian yang menitikberatkan pada aspek fungsional dari suatu sistem. Syarat dimulainya fase ini adalah pengujian di fase SIT sudah selesai. Aspek fungsional yang di uji diantaranya terkait *user-interface*, fitur-fitur yang bisa secara langsung di akses oleh user, dan laporan manajemen. Pengujian pada fase ini sepenuhnya dilakukan oleh user yang dibantu oleh tester. Tujuan dari fase ini selain melihat sistem yang di uji dari sudut pandang fungsional adalah sebagao dan peletak dasar sebelum sistem di implementasi ke pengguna yang sebenarnya. Data yang digunakan selama dilakukan pengujian dalam fase ini adalah data simulasi, tapi kadang data yang digunakan adalah data asli sesuai dengan keinginan dan ijin dari user.

Environment production, atau yang disebut PR stage (fase *prouduction review*), adalah pengujian yang dilakukan setelah sistem secara resmi di migrasi ke ranah produksi. Pengujian di fase ini penuh resiko, oleh karena itu pengujian biasanya dilakukan seminim dan seefektif mungkin dengan menitikberatkan pada fungsi-fungsi utama saja. Resiko yang terjadi (terutama kalau pengujian sistem perbankan) adalah data nasabah yang riil, bukan data simulasi seperti yang digunakan pada fase-fase sebelumnya. Pengujian pada fase ini dilakukan oleh user yang berkoordinasi dengan team *operation* dan tester.

Membagi pengujian dalam multi-environment juga memudahkan manajemen dalam melihat benar atau tidak nya proses SDLC berjalan dalam proses pengembangan sistem. Pada tahap SIT stage tidak ditemukan kesalahan (bug), tapi bug ditemukan kembali di UAT stage, dan bahkan seringkali ditemukan kembali di PR stage. Hal ini membuat proses testing harus dilakukan dengan perencanaan, mekanisme dan prosedur yang baik.

Test planning dan Test Strategy

Test plan dan *test strategy* mendeskripsikan kapan dan bagaimana proses pengujian akan dilakukan. Test plan juga memberikan informasi mengenai testing, background testing, tujan dan resiko testing, dan juga mekanisme bagaimana fungsi-fungsi bisnis dalam aplikasi akan di uji. Hal-hal yang penting dalam test plan adalah:

1. *User requirement*
2. *Project scope*
4. Tipe testing
5. *Test design*
6. *Test data management*
7. *Testing tools*

User Requirement

User requirement dalam testing dibagi dalam 2 domain, yaitu functional requirement dan non-functional requirement. Walaupun dalam SDLC posisi pengujian terletak pada fase bawah, akan tetapi secara praktis keterlibatan user, tester, dan *quality assurance* di mulai di saat fase pertama pertama kali di gelar. Pengetahuan dan pengalaman seorang tester sangat dibutuhkan dalam fase *user requirement*. Selain dapat menjadi dasar pembuatan skenario testing dan *test case*, fase ini juga mendorong tester dan QA untuk menyelami lebih dalam perilaku dari sistem dari sisi kebutuhan fungsional serta menjadi *engagement ring* / mediator antara *business analyst*, developer dan user.

Project Scope

Project scope dilihat dari berbagai parameter. Mulai dari jadwal testing, orang-orang yang bertanggung jawab atau terlibat dalam testing, hingga besaran dan ukuran dari testing. Ukuran dan besaran dari testing menjelaskan berbagai dukungan yang digunakan dalam testing, seperti penggunaan perangkat (komputer, laptop, tablet, telepon seluler, dan lainnya), penggunaan sistem operasi (linux, windows, android, atau ios), penggunaan *platform*, penggunaan database, dan dukungan lainnya yang harus ditetapkan untuk menjaga besaran dan ukuran pengujian berada dalam koridor yang benar. Penetapan besaran dan ukuran ini sangat membantu manajemen dalam menghitung budget, sumber daya yang digunakan dan jadwal pengujian.

Tipe testing

Beberapa tipe testing yang paling lazim dilakukan oleh perusahaan pengembangan sistem adalah, *full cycle* testing atau *integration testing*,

regression testing, *automation testing*, *smoke testing*, dan *penetration testing*.

Full cycle testing atau integration testing adalah suatu kegiatan dengan menjalankan seluruh pengujian fung-sional berdasarkan skenario testing seluruh yang sepenuhnya disadur dari dokumen *user requirement*. Testing ini dilakukan pertama kali oleh tester pada *environment development* dalam SIT stage (fase *system integration testing*). Seluruh user requirement yang sudah di terjemahkan ke dalam skenario testing harus di uji sepenuhnya secara detail. *Integration test* yang berada dalam *full cycle testing* ini menitikberatkan pada semua kegiatan testing. Baik *functional test* atau *blackbox test* yang dilakukan oleh test tester, dan *unit test* atau *whitebox test* yang di kerjakan oleh developer.

Regression testing adalah pengujian yang dilakukan terhadap suatu sistem yang sebelumnya sudah berjalan sebagai akibat adanya perubahan, penambahan, ataupun perbaikan modul. Regression testing sering juga disebut dengan pengujian ulang, dilakukan sebagai antisipasi jika segala perubahan atau perbaikan yang terjadi mempunyai dampak yang serius dan signifikan terhadap suatu sistem. Regression harus dilakukan pada semua environment, mulai dari SIT *stage*, UAT *stage*, hingga PR *stage*.

Automation testing adalah serangkaian kegiatan test yang dilakukan menggunakan secara non manual dan menggunakan aplikasi tertentu. Dengan kata lain, mulai dari pembuatan skenario testing, hingga eksekusi testing dilakukan secara otomatis. Tidak seluruh aplikasi dapat dilakukan test secara otomatis, banyak yang masih harus dilakukan secara manual. Beberapa contoh fungsi yang sering dilakukan dengan automation testing diantaranya adalah login, registration form, koneksi database, dan beberapa pengecekan GUI lainnya. Testing pada fungsi yang memuat transaksi keuangan ataupun yang berhubungan dengan pihak ketiga masih dilakukan secara manual dan semi otomatis. *Automation test* biasa dilakukan dengan beberapa alasan antara lain, *requirement* secara konstan tidak atau jarang sekali berubah, beberapa projek dengan *behavior* yang sama, dan menguji *load* atau kinerja sistem yang biasanya hanya dilakukan oleh perusahaan sebanyak sekali atau dua kali dalam setahun.

Smoke test merupakan test cepat terhadap fitur-fitur yang penting dalam suatu aplikasi. *Smoke test* dilakukan dalam hitungan jam, fokus pada GUI dan transaksi yang kritikal. *Smoke test*, atau kadang disebut dengan *sanitation test* biasa dilakukan setelah suatu fitur di *deploy*/migrasi ke *environment* baru. Para tester melakukan ini pada saat pre-UAT, pre-Production, dan sebelum *regression test*. *Smoke test* wajib dilakukan untuk menghindari *defect* atau *bug* yang bersifat minor akan tapi sering menjadi hambatan dalam proses pengembangan sistem.

Penetration testing, yang sering disingkat *pentest* adalah suatu kegiatan dimana tester bekerjasama dengan seorang *security analyst* mencoba mensimulasikan serangan yang bisa dilakukan kedalam suatu jaringan organisasi/perusahaan tertentu untuk menemukan kelemahan yang ada pada sistem jaringan tersebut. *Pentest* sering dilakukan pada aplikasi yang banyak memuat transaksi keuangan dan terkait kerahasiaan nasabah. Aplikasi perbankan atau aplikasi produk keuangan lainnya rutin dilakukan pentest untuk menghindari serangan-serangan yang bisa menyebabkan banyak kerugian secara finansial terhadap nasabah mereka.

Test Design

Test design merupakan aktivitas membuat dan menuliskan skenario testing dalam suatu pengembangan sistem. Pembuatan dan penulisan skenario testing tidaklah mudah. Seseorang harus mengetahui dan mempunyai syarat sebagai berikut sebelum menuliskan skenario.

1. Mengetahui proses bisnis sesuai dengan area yang akan di uji
2. Memahami sejarah dan perilaku dari aplikasi yang akan di uji
3. Paham akan fungsi-fungsi atau fitur yang terdapat dalam sistem
4. Menguasai pengetahuan dari testing itu sendiri
5. Dan *awareness* terhadap mekanisme testing ataupun scope dari testing.

Test data management

Test data adalah segala informasi yang digunakan sebagai input/masukan dalam melakukan testing. Test data bisa bisa berupa data statis atau data transaksional seperti kode user name, momor identitas, nomor rekening bank, nomor kartu kredit dan sebagainya.

Test data management, adalah bagaimana mengelola data-data agar sesuai dengan kebutuhan testing. Baik data yang harus sesuai dengan format yang diinginkan oleh programmer atau user,

dan *data scramble* yang mewakili data nasabah tapi dipastikan kerahasiaannya terjaga.

Test data management merupakan bagian dari testing yang penting, akan tetapi sering terlewatkan. Melakukan testing tanpa test data yang berkualitas akan mengakibatkan hasil test yang diragukan kualitasnya. Masalah yang sering timbul adalah, tester mempunyai kewajiban membuat analisis pada berbagai test data untuk keperluan *negative testing*, tapi data tersebut sering tidak dapat disediakan oleh programmer. Test data management harus di komunikasikan dengan bagian developer sejak awal siklus pengembangan sistem dimulai.

Testing tools

Alat bantu testing, sering disebut dengan *testing tools*, *testing manager*, atau *test director*, adalah suatu aplikasi yang berguna untuk membantu jalannya testing. Aplikasi testing ini tidak hanya digunakan untuk membantu beberapa testing dari manual menjadi otomatis. Aplikasi ini juga digunakan sebagai *issue tracking* dan *project management*. Mengingat semakin besar aktivitas testing dan semakin besar kapasitas hubungan kerja antara testing, developer, project management dan user, maka tools seperti *Jira*, *redmine* atau HP *quality center* menjadi pilihan penting untuk memayungi seluruh kegiatan testing mulai dari review dokumen, pembuatan skenario testing sampai ke eksekusi testing bersama-sama dengan divisi terkait lainnya.

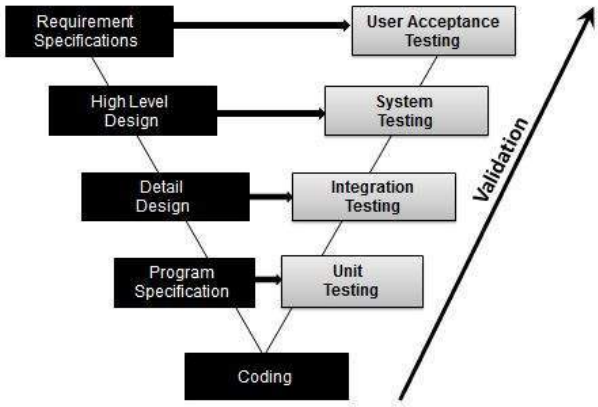
Deskripsi Pekerjaan Quality Assurance dan tester

Tester dalam beberapa kondsi kadang di-analogikan seperti warung kopi. Artinya, ada aplikasi, selesai dibuat, silahkan dan segera di test. Mengingat dalam SDLC testing berada di posisi akhir dan tidak berkesan strategis menyebabkan pekerjaan testing tidak sepenting pekerjaan seorang database administrator, system analis dan programer. Padahal tidak demikian. Eksekusi memang di akhir, tapi formulasi dan strategi berada di awal. Anisisis terhadap dokumen bisnis, pembuatan skenario testing dan memahami scope adalah bagian penting yang harus dilakukan tester sejak awal suatu proyek dimulai. Berikut adalah beberapa tugas penting dari seorang QA/tester:

1. Identifikasi kebutuhan bisnis, *review* dokumen bisnis
2. Identifikasi timeline, *scope*, test data, *environment*, dan *impact analysis* (*Test Plan*)
3. Membuat skenario testing / test cases
4. Bekerjasama dengan internal (developer, *project manager*) dan eksternal (user, *product owner*)
6. Eksekusi testing
7. Membuat laporan dan memberikan rekomendasi ke manajemen.

Test validation

Test validation merupakan proses evaluasi sistem atau perangkat lunak baik di saat proses pengembangan dimulai, proses pengembangan berjalan, hingga proses akhir pengembangan untuk mengawal kegiatan testing sesuai dengan tujuan yang telah ditetapkan. Seorang *Quality assurance* harus memastikan suatu produk tidak hanya berkualitas secara fisik (delivered), tetapi juga berkualitas dalam proses dan prosedur (gambar 1). Test validation berarti tester harus memastikan dan memonitor bahwa dirinya sendiri dan stakeholder lainnya (programer, project manajer, dan departemen terkait lainnya) telah melakukan pekerjaan sesuai dengan prosedur dan mekanisme yang ada.



Gambar 1: Test validation. Sumber: https://www.tutorialspoint.com

Kesimpulan

Testing dalam siklus pengembangan sistem tidak hanya mencari kesalahan dari perangkat lunak, testing mencakup pengetahuan terhadap perangkat lunak itu sendiri. Pengetahuan terhadap perilaku sistem, pemahaman terhadap mekanisme kerja pengembangan perangkat lunak, prosedural dan proses bisnis adalah bagian penting dari kegiatan testing dalam daur hidup pengembangan sistem. Testing tidak hanya bicara *do the right things* (melakukan hal yang benar sesuai tujuan: menguji perangkat lunak, mencari *bug*, dan melaporkan). Testing harus memberikan perspektif do the things right (sudahkah kita melakukan sesuatu dengan benar?). Artinya, selain tester harus paham terhadap seluruh dokumen fungsional, mengerti scope dari proyek sistem informasi, tester harus memastikan, mengawal dan memonitor bahwa seluruh pekerjaan yang berhubungan dengan eksekusi testing baik yang berhubungan dengan developer, project manager dan user harus berada dalam koridor dan proses yang telah disepakati.

Daftar pustaka

Sommerville, Ian. Software Engineering 9th edition, Pearson, 2011

Perry, William E. Effective method for software testing, Wiley, 2006

Pressman, Roger. Software engineering: A practitioner's approach, McGraw Hill, 2005

Article, IEEE – Standard 1012 – 1998, standard for software verification and validation

Case study: Ogilvy dan Bullseye Digital, berdasarkan dokumen dan riset, 2015

http://softwaretestingfundamentals.com, 2016

http://softwaretestingessentials.com, 2016

https://www.altassian.com/software/jira, 2017

www.redmine.org, 2017

https://www.tutorialspoint.com