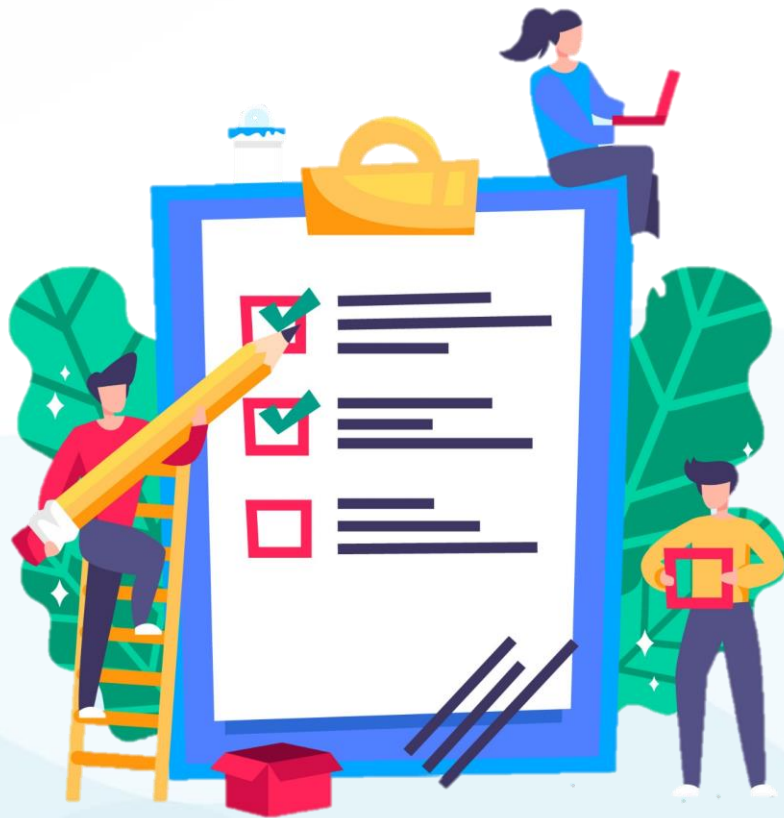




Program Studi Teknik Informatika

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

MATA KULIAH
SOFTWARE QUALITY & TESTING



Pengujian Skenario

Apakah Skenario Pengujian ?

- Skenario pengujian didefinisikan sebagai fungsi-fungsi apa saja yang dapat diuji. Skenario pengujian juga disebut Test Condition (Kondisi Pengujian) atau Test Possibility (Kemungkinan Pengujian)
- Sebagai tester/ penguji, memposisikan diri pada posisi end user, dan menggambarkan skenario dunia nyata dan menggunakan kasus-kasus dari Application Under Test (kasus pada aplikasinya).

Apakah Pengujian Skenario ?

- Pengujian skenario merupakan varian dari pengujian perangkat lunak dimana skenario digunakan untuk pengujian.
- Skenario membantu dalam memudahkan pengujian dari system yang kompleks.

Mengapa membuat Skenario Pengujian ?

Alasan membuat skenario pengujian, adalah :

- Pembuatan skenario pengujian untuk memastikan cakupan pengujian lengkap
- Skenario pengujian dapat disetujui oleh berbagai pemangku kepentingan (stakeholder) seperti analis bisnis, pengembang, pelanggan untuk memastikan Application Under Test (aplikasi yang diuji) diuji secara menyeluruh.
- Hal ini memastikan bahwa perangkat lunak berfungsi untuk kasus-kasus yang paling umum.
- Skenario pengujian merupakan alat cepat untuk menentukan upaya kerja pengujian dan pembuatan proposal yang sesuai untuk klien atau mengatur tenaga kerja.
- Membantu menentukan transaksi *end-to-end* yang paling penting atau penggunaan sebenarnya dari aplikasi perangkat lunak.

Kapan Skenario Pengujian Tidak diperlukan ?

- Aplikasi yang diuji rumit, tidak stabil, dan ada krisis waktu dalam proyek.
- Proyek yang mengikuti Metodologi Agile seperti Scrum, Kanban mungkin tidak membuat scenario pengujian.
- Skenario pengujian mungkin tidak dibuat untuk perbaikan bug atau pengujian regresi (Pengujian regresi adalah proses pengujian produk perangkat lunak setelah perubahan dilakukan untuk memastikan bahwa tidak ada bug baru yang diperkenalkan sebagai akibat dari perubahan tersebut). Dalam kasus itu skenario pengujian mungkin sudah berat didokumentasi dalam siklus pengujian sebelumnya.

Lima tahapan menulis Skenario Pengujian





1. Baca dokumen kebutuhan seperti BRS, SRS, FRS, dari system yang diuji (System Under Test – SUT). Dapat mengacu use cases, buku, manual atau lainnya dari aplikasi yang diuji.
2. Untuk setiap kebutuhan, gambarkan aksi dan tujuan dari user yang mungkin. Tentukan aspek teknis kebutuhan. Pastikan skenario kemungkinan dari penyalahgunaan system dan evaluasi pengguna dengan pola pikir hackers.
3. Setelah membaca dokumen kebutuhan dan melakukan analisis jatuh tempo, buat daftar skenario pengujian yang berbeda yang memverifikasi setiap fitur perangkat lunak.
4. Setelah membuat daftar semua kemungkinan skenario pengujian, Matrix Pelacakan dibuat untuk memverifikasi bahwa setiap dan semua kebutuhan memiliki skenario pengujian yang sesuai.
5. Skenario yang dibuat ditinjau ulang oleh supervisor. Nantinya, mereka juga direview oleh stakeholder dalam proyek tersebut.

Tips membuat Skenario Pengujian


- Setiap skenario pengujian harus dikaitkan dengan minimal satu kebutuhan atau user story yang sesuai Metodologi Proyek.
- Sebelum membuat skenario pengujian yang memverifikasi beberapa kebutuhan sekaligus, pastikan anda memiliki skenario yang memeriksa bahwa kebutuhan terpisah.
- Hindari membuat rentang beberapa kebutuhan skenario yang terlalu rumit
- Jumlah skenario mungkin besar, dan mahal untuk dijalankan semuanya. Maka skenario pengujian hanya dijalankan yang dipilih berdasarkan prioritas pelanggan

Contoh 1 : Skenario Pengujian untuk Aplikasi e- Commerce



ZALORA WANITA PRIA LUXURY ANAK HEALTH & BEAUTY MAINAN & HOBI Offers Up to 50%    

Masuk ke akun Anda

 Lanjutkan dengan Facebook

Kami tidak akan posting atas nama Anda atau membagikan informasi apapun tanpa persetujuan Anda.

_____ atau _____

Alamat Email

Password

 [Tampilkan](#)

[Lupa password?](#)

—

Z

Untuk membantu memahami perbedaan Skenario pengujian dan Kasus Uji, kasus uji khusus untuk scenario pengujian akan di :

1. Cek perilaku system Ketika id email dan password yang valid dimasukan.
2. Cek perilaku system Ketika id email invalid dan password valid dimasukan.
3. Cek perilaku system Ketika id email valid dan password invalid dimasukan.
4. Cek perilaku system Ketika id email dan password invalid dimasukan.
5. Cek perilaku system Ketika id email dan password dibiarkan kosong masuk.
6. Cek lupa kata sandi berfungsi seperti yang diharapkan.
7. Cek perilaku system ketika nomor telp valid/ tidak valid dimasuka .
8. Cek perilaku system Ketika “ biarkan saya tetap ditangani” dicentang.

Cek Fungsi Pencarian



Cek Fungsi Pencarian

Cek Halaman Deskripsi Produk



Cek Fungsi Pembayaran

☐ Credit card



☐ Debit card

Choose an Option ▾

☐ Net Banking

Choose an Option ▾

☐ BHIM UPI [What is BHIM UPI?](#)

☐ EMI Unavailable [Why?](#)

☐ Pay on Delivery (POD)

We also accept Credit/ Debit cards on delivery, subject to availability of the payment dev

[Know more.](#)

Cek History Pesanan

AKUN SAYA

Akun Saya

Wallet

Cashback

ZNOW Saya

Lacak Pesanan

Penukaran & Pengembalian

Konfirmasi Transfer

Preferensi Email

Wishlist

Lacak Pesanan

Nomor Order

Dipesan Pada 2020-10-17

LIHAT PEMESANAN

Ukuran L

Rp 629.930

Ukuran L

Rp 629.930

Terkirim

Status Diperbarui: 2020-10-22 14:05

LACAK PESANAN

Terlepas dari lima skenario sebelumnya, berikut adalah daftar dari skenario lainnya, yaitu :

- Periksa perilaku Home Page untuk pelanggan yang kembali.
- Cek halaman kategori/ produk
- Cek halaman kontak/ layanan pelanggan
- Cek halaman Penawaran Harian



Contoh 2 : Skenario Pengujian untuk Aplikasi Situs Bank

Skenario Pengujian 1 : Cek Fungsi Login dan Keaslian

Skenario Pengujian 2 : Cek Transfer Uang dapat dilakukan

Skenario Pengujian 3 : Cek Laporan Rekening dapat dilihat

Skenario Pengujian 4 : Cek Deposit Tetap/ Deposit berulang
dapat dibuat



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

Bagaimana menulis kasus Uji ?

Bagaimana Menulis Kasus Uji ?

- Kasus Uji adalah serangkaian Tindakan yang dijalankan untuk memverifikasi fitur atau fungsi tertentu dari aplikasi perangkat lunak. Kasus uji adalah komponen yang paling diperlukan dari Siklus Hidup Pengujian Perangkat lunak yang membantu memvalidasi AUT (Application Under Test)

Skenario Pengujian VS Kasus Uji

- Skenario Pengujian agak tidak jelas dan mencakup berbagai kemungkinan. Pengujian adalah tentang semua yang menjadi sangat spesifik.
- Untuk Skenario Pengujian : Cek fungsi Login disana banyak kemungkinan kasus uji, yaitu :
 - Kasus Uji 1 : Cek hasil dengan memasukkan Id user dan password valid
 - Kasus Uji 2 : Cek hasil dengan Id user dan password Invalid
 - Kasus Uji 3 : Cek respon Ketika id user kosong dan tombol Login ditekan, dan masih banyak lagi.

Membuat Kasus Uji

- Kasus Uji untuk skenario : Cek Fungsi Login



The image shows a login interface with two tabs: 'Pelanggan baru' and 'Pelanggan lama'. The 'Masuk ke akun Anda' section includes a 'Lanjutkan dengan Facebook' button, a disclaimer, and fields for 'Alamat Email' and 'Password'. A red rounded rectangle with the text 'Cek Fungsi Login' is overlaid on the right side of the form.

Tahap 1 : Simple kasus uji untuk scenario

Kasus Uji	Deskripsi Kasus Uji
1	Cek respon Ketika email dan password valid dimasukkan

Tahap 2 : Untuk menjalankan kasus uji, kita memerlukan Data Uji

Kasus Uji	Deskripsi Kasus Uji	Data Uji
1	Cek respon Ketika email dan password valid dimasukkan	Email:BudiSqt@dsn.dinus.ac.id

Data uji perlu didokumentasi, karena mengidentifikasi data uji bisa memakan waktu dan terkadang perlu membuat data uji lagi

- Step 3. menjalankan kasus uji, seorang penguji perlu melakukan serangkaian aksi tertentu di AUT (Application Under Test)
- Dokumen ini seperti yang ada di bawah :

Test Case	Test Case Description	Test Steps	Test Data
1	Cek respon ketika email dan password valid dimasukan	1. Masukan alamat email 2. Masukan Password 3. Tekan Sign in	Email : BudiSqt@dsn.dinus.ac.id Password : D21*Ona^iz

- Seringkali langkah pengujian tidak sesederhana di atas, maka diperlukan dokumentasi. Selain itu juga yang melakukan kasus uji dapat melakukan aktiitas lain.
- Karyawan barupun dapat melaksanakan kasus uji ini.
- langkah-langkah yang terdokumentasi akan membantu penguji dan juga memfasilitasi reiew yang dilakukan oleh stekholder.

- Step 4. Tujuan dari kasus uji adalah untuk memeriksa perilaku AUT untuk sebuah hasil yang diharapkan. Hal ini perlu didokumentasi sbb :

Test Case	Test Case Description	Test Data	Expected Result
1	Cek respon Ketika email dan password valid dimasukan	Email : BudiSqt@dsn.dinus.ac.id Password : D21*Ona^iz	Login seharusnya berhasil

- Selama waktu eksekusi pengujian, tester akan memeriksa hasil yang diharapkan terhadap hasil yang actual dan menetapkan status berhasil atau gagal

Test Case	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Cek respon Ketika email dan password valid dimasukan	Email : BudiSqt@dsn.dinus.ac.id Password : D21*Ona^iz	Login seharusnya berhasil	Login berhasil	Pass

- Step 5. Terlepas dari kasus uji, mungkin memiliki bidang seperti, Pre-condition dimana yang menentukan hal-hal yang harus ada sebelum pengujian dilakukan.
- Untuk kasus e-commerce di atas, prasyaratnya adalah memiliki browser agar memiliki akses ke situs syang diuji.
- Kasus uji juga termasuk Post- Condition yang menentukan apa saja yang berlaku setelah kasus uji selesai.
- Untuk contoh, kondisi pasca adalah waktu dan tanggal login disimpan pada database.

Format Standar Kasus Uji

- Di bawah ini adalah format standar Kasus Uji Login

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Cek login pelanggan dengan data valid	<ol style="list-style-type: none">1. Go to site http://dinus.ac.id2. Enter UserId3. Enter password4. Tekan Submit	UserId = NinaDinus Password = passDinus	User harus masuk ke dalam aplikasi	Sesuai yang diharapkan	Pass
TU02	Cek login pelanggan dengan data invalid	<ol style="list-style-type: none">1. Go to site http://dinus.ac.id2. Enter UserId3. Enter password4. Tekan Submit	UserId = NinaDinus Password = passNinus	User seharusnya tidak bisa login ke aplikasi	Sesuai seperti yang diharapkan	Pass

Informasi berikut penting Ketika Menyusun Kasus Uji :

1. Deskripsi kebutuhan apa yang sedang diuji.
2. Penjelasan bagaimana system akan diuji
3. Pengaturan pengujian seperti versi, software, file data, system operasi, hardware, akses keamanan, tanggal fisik atau logic, waktu, prasyarat seperti pengujian lain dan informasi pengaturan apa saja yang berkaitan dengan persyaratan yang diuji dari aplikasi yang sedang diuji.
4. Input dan output atau aksi dan hasil yang diharapkan.
5. Bukti atau lampiran apapun
6. Gunakan Bahasa kasus aktif
7. Kasus uji seharusnya tidak lebih dari 15 langkah
8. Skrip pengujian otomatis diberi komentar dengan input, tujuan dan hasil yang diharapkan
9. Mempersiapkan penawaran alternatif untuk tes prasyarat.
10. Dengan pengujian lain, hal tersebut harus menjadi urutan scenario bisnis yang salah.

Praktek terbaik untuk penulisan contoh kasus uji yang baik

1. Kasus uji harus sederhana dan trasparan
2. Buat kasus uji dengan pengguna akhir dalam pikiran
3. Hindari pengulangan kasus uji.
4. Jangan Berasumsi
5. Pastikan Cakupannya 100%
6. Kasus Uji harus dapat diidentifikasi
7. Menerapkan Teknik Pengujian (BVA, EP, State Transition Technique, Error Guessing Technique)
8. Membersihkan Diri sendiri (aplikasinya)
9. Berulang dan berdiri sendiri
10. Review Sejawat

Tool Pengelolaan Kasus Uji

- Tools pengelolaan pengujian merupakan alat otomatis yang membantu untuk mengelola dan memaintain kasus uji.
- Fitur utama dari tool pengelolaan kasus uji meliputi :
 1. Untuk pendokumentasian Kasus Uji : dengan alat, kita dapat mempercepat pembuatan kasus uji dengan menggunakan template
 2. Menjalankan Kasus Uji dan Rekam hasilnya : Kasus uji dapat dieksekusi melalui alat dan hasil yang diperoleh dapat mudah direkam
 3. Otomatisasikan Pelacakan Kesalahan/ cacat : tes yang gagal secara otomatis ditautkan ke pelacak Bug, yang pada gilirannya dapat ditugaskan ke pengembang dan dapat dilacak dengan pemberitahuan email.
 4. Penelusuran : Kebutuhan, kasus uji, eksekusi kasus uji semuanya saling berkaitan melalui alat, dan setiap kasus dapat dilacak satu sama lain untuk memeriksa cakupan pengujian.

Tool Pengelolaan Kasus Uji

5. Melindungi Kasus Uji :

- kasus uji harus dapat digunakan Kembali dan harus dilindungi dari hilang atau rusak karena minimnya pengendali control.
- Alat pengelolaan kasus uji menawarkan fitur seperti :
 - a. Konvensi penamaan dan penomoran
 - b. Pembuatan versi
 - c. Penyimpanan read-only
 - d. Akses terkontrol
 - e. Pencadangan di luar situs

Teknik Pengujian Perangkat Lunak

- Teknik pengujian perangkat lunak membantu dalam merancang kasus uji yang lebih baik, Ketika pengujian menyeluruh tidak mungkin dilakukan.
- Teknik pengujian manual membantu mengurangi jumlah kasus uji yang akan dijalankan dan meningkatkan cakupan pengujian.
- Hal ini membantu mengidentifikasi kondisi pengujian yang sulit untuk dikenali.

Teknik Pengujian Perangkat Lunak - Boundary Value Analysis

- Analisis Nilai Batas adalah berbasis pengujian pada nilai-nilai batas antara partisi, mencakup maksimum, minimum, di dalam, atau di luar batas, nilai tipikal dan nilai kesalahan.
- Secara umum terlihat bahwa sejumlah besar kesalahan terjadi di batas dari nilai input yang ditentukan daripada pusatnya.
- Teknik pengujian kotak hitam melengkapi kesetaraan partisi.
- Teknik pengujian perangkat lunak ini berdasarkan prinsip bahwa, jika suatu sistem bekerja dengan baik untuk nilai-nilai tertentu, maka akan berhasil sangat baik untuk semua nilai yang berada di antara dua batas tersebut

Apakah Requirement Traceability Matrix (RTM) ?

Traceability matrix adalah dokumen yang berhubungan dengan beberapa dokumen dua baseline yang membutuhkan hubungan banyak ke banyak untuk diperiksa kelengkapan dari hubungan tersebut.

Ini digunakan untuk melacak kebutuhan dan untuk memeriksa kebutuhan proyek saat ini yang terpenuhi.

Apakah Requirement Traceability Matrix (RTM) ?

RTM mencakup semua kebutuhan yang diusulkan oleh client atau tim pengembang perangkat lunak dan mereka melacak dalam satu dokumen yang dikirim pada akhir siklus hidup pengembangan perangkat lunak.

Dengan kata lain, RTM adalah dokumen yang memetakan dan melacak kebutuhan user dengan kasus uji.

Tujuan utama RTM adalah untuk melihat bahwa semua kasus uji tercover sehingga tidak ada fungsi yang harus terlewat saat melakukan pengujian perangkat lunak.

Teknik Pengujian Perangkat Lunak - Apakah Pengujian Statik ?

- Pengujian statik didefinisikan sebagai Teknik pengujian perangkat lunak yang dapat memeriksa cacat pada perangkat lunak tanpa benar-benar menjalankannya.
- Pengujian dinamis adalah pengujian yang memeriksa aplikasi Ketika program dijalankan.
- Pengujian statis dilakukan untuk menghindari kesalahan pada tahap awal pengembangan karena lebih mudah Ketika menemukan sumber kegagalan dan kegagalannya itu sendiri.
- Pengujian statis membantu menemukan kesalahan yang mungkin tidak ditemukan oleh pengujian dinamis.

Apakah Review Pengujian ?

- Review dalam pengujian statis adalah proses atau pertemuan yang diadakan untuk menemukan cacat yang berpotensi dalam perancangan program apapun.
- Selain itu juga review digunakan agar semua anggota tim mengetahui tentang kemajuan proyek, dengan memahami bersama harapannya pemikiran yang beragam dapat menghasilkan saran yang sangat bagus, dokumen langsung diperiksa dan perbedaan dipilah-pilah.

Menguji lingkungan untuk Pengujian Perangkat Lunak

- Lingkungan pengujian adalah pengaturan software dan hardware untuk tim penguji dalam melaksanakan kasus uji. Pengujian ini mendukung eksekusi pengujian dengan hardware dan software, dan jaringan yang dikonfigurasi.
- Tempat pengujian dan lingkungan pengujian dikonfigurasi sesuai kebutuhan aplikasi yang diuji (AUT). Pada beberapa kesempatan, tempat pengujian dapat dipadukan/ dikombinasikan menjadi lingkungan pengujian dan data pengujian yang dioperasikan.
- Menyiapkan lingkungan pengujian yang tepat memastikan keberhasilan pengujian perangkat lunak. Setiap kekurangan dalam proses ini dapat menyebabkan biaya dan tambahan waktu bagi client.

Menguji lingkungan untuk Pengujian Perangkat Lunak

Area utama yang disiapkan dalam Lingkungan Pengujian :

- Sistem dan aplikasi
- Data uji
- Server database
- Lingkungan yang berjalan front-end
- Sistem operasi klien
- Browser
- Perangkat keras termasuk server sistem Operasi
- Jaringan
- Dokumentasi yang diperlukan

Test Data Generation (Apa, bagaimana, contoh, dan tools)

- Pengujian adalah proses yang menghasilkan dan membutuhkan data dalam jumlah yang sangat besar.
- Data yang digunakan dalam pengujian menggambarkan kondisi awal untuk pengujian dan mewakili media yang digunakan untuk penguji dalam mempengaruhi perangkat lunak.
 - Apa itu Data Uji ?
 - Mengapa menggunakan Data Uji ?

Proses Mengelola Cacat dalam Pengujian Perangkat Lunak

- Bug adalah konsekuensi atau hasil dari salah koding
- Cacat adalah variasi atau penyimpangan dari kebutuhan bisnis aslinya
- Kedua istilah di atas memiliki perbedaan yang sangat tipis.
- Dalam industry keduanya merupakan kesalahan yang harus diperbaiki dan perubahannya digunakan oleh beberapa tim penguji.
- Ketika seorang penguji menjalankan kasus uji, dia mungkin menemukan hasil uji yang bertentangan dengan hasil yang diharapkan.
- Penyimpangan dalam hasil uji disebut sebagai cacat perangkat lunak.
- Cacat ini atau penyimpangan dirujuk dengan nama yang berbeda di organisasi yang berbeda seperti masalah , bug, atau insiden.

Terimakasih

Ada Pertanyaan?



Referensi

- Beizer-90: Boris Beizer, Software Testing Techniques, Van Nostrand Reinhold., New York, 1990, ISBN 0-442-20672-0.
- GSAM: Department of the Air Force Software Technology Support Center, Guidelines for Successful Acquisition and Management of SoftwareIntensive Systems: Weapon Systems Command and Control Systems Management Information Systems, Version 3.0, May 2000, Hill Air Force Base, Utah 84056-5205, <https://www.stsc.hill.af.mil>
- IEEE-610: IEEE Standards Software Engineering, Volume 1, IEEE Standard Glossary of Software Engineering Terminology, IEEE Std. 610-1990 , The Institute of Electrical and Electronics Engineers, 1999, ISBN 0- 7381-1559-2.
- McCabe-82: Thomas J. McCabe, Structured Testing: A Software Testing Methodology Using the Cyclomatic Complexity Metric, NBS Special Publication, National Bureau of Standards, 1982.
- Pressman-01: Roger Pressman, Software Engineering, A practitioner's Approach, Fifth Edition, McGraw Hill, Boston, 2001, ISBN 0-07-365578-3.
- On-Line Resources (note: The use of all on-line resources is subject to applicable copyright laws and may change with time and publication status. The net locations given below should also be considered dynamic, but are accurate at the completion of this paper.)
- Unpublished notes on basis path testing are available at:
 - Check Yang <http://www.chekyang.com>
 - Dr. A.J. Sobey http://louisa.levels.unisa.edu.au/se1/testing-notes/test01_3.htm
 - Joseph Poole NISTIR 5737 - <http://hiss.nist.gov/publications/nistir5737/>