

#06

Test Case Development



AIK21361 (3 sks)

UJI PERANGKAT LUNAK

Nurdin Bahtiar, S.Si., M.T.

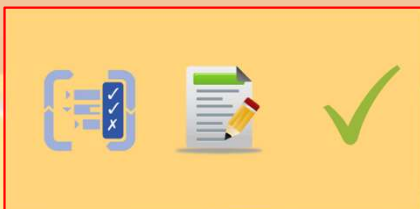
-

1. Test Case Development



Test case merupakan suatu prosedur detail yang secara penuh menguji suatu *feature* atau aspek dari *feature*.

Kalau *test plan* menjelaskan apa yang diuji, sedangkan *test case* menjelaskan bagaimana melakukan suatu pengujian partikular.



Test case harus dikembangkan untuk setiap daftar pengujian yang ada dalam *test plan*.

1. Test Case Development



Petunjuk umum:

- ☐ Sebagai penguji, jalan terbaik untuk menggambarkan kepatuhan perangkat lunak terhadap kebutuhannya adalah dengan mendesain test case efektif yang menyajikan pengujian yang melingkupi unit.
- ☐ Desain test case yang bervariasi membuat penguji dapat mengembangkan test case yang efektif.

1. Test Case Development



Panduan dalam membantu merancang test case:

1

Tujuan dari setiap test case adalah untuk menjalankan pengujian dengan cara yang se-simple mungkin.

2

Konsentrasilah / fokus sejak awal pada pengujian yang positif, misalnya test case harus dapat menunjukkan bahwa yang dilakukan software adalah apa yang seharusnya dilakukan.

3

Test case yang ada seharusnya semakin meningkat dan test case selanjutnya harus didisain untuk menunjukkan bahwa yg dilakukan software bukanlah sesuatu yg tidak dispesifikasikan.

4

Test case seharusnya didisain untuk ditujukan pada hal-hal seperti performa, kebutuhan kenyamanan, dan keamanan.

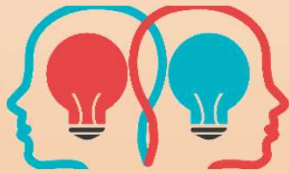
5

Test case lebih jauh dapat ditambahkan pada spesifikasi pengujian unit untuk mencapai area objektif yang lebih spesifik.

2. Defect Tracking



Defect is the variance from a desired product attribute (it can be a wrong, missing, or extra data).



It can be of two types:

- ☐ Defect from the product
- ☐ A variance from customer / user expectations.



Macam-macam defect:

- ☐ Security Defects
- ☐ Data Quality / Database Defects
- ☐ Critical Functionality Defects
- ☐ Functionality Defects
- ☐ User Interface Defects

2. Defect Tracking



Security Defects:

Umumnya melibatkan penanganan yang tidak benar akan data yang dikirim oleh pengguna dalam aplikasi. Ini merupakan *defect* yang paling penting dan prioritas tinggi untuk dibenahi.

Misalnya:

- ☐ Authentication: Accepting an invalid username / password
- ☐ Authorization: Accessibility to pages though permission not given.



Data Quality / Database Defects:

Berhubungan dengan penanganan data dalam basis data yang tidak benar.

Misalnya:

- ☐ Values not deleted / inserted into the database properly
- ☐ Improper / wrong / null values inserted in place of the actual values.

2. Defect Tracking



Critical Functionality Defects:

Bug yang menghambat fungsionalitas aplikasi secara kritis.

Misalnya:

- ☐ Exceptions



Functionality Defects:

Defect yang mempengaruhi fungsionalitas aplikasi.

Misalnya:

- ☐ Buttons like Save, Delete, Cancel, not performing their intended functions



User Interface Defects:

Berhubungan dengan permasalahan terkait UI yang biasanya kurang dipertimbangkan.

Misalnya:

- ☐ Improper error / warning / UI messages or Spelling mistakes

3. Type of Test Report



Test reporting meliputi empat tipe dokumen:

1. **A Test Item Transmittal Report**

Mengidentifikasi item test yang akan dilakukan pengujian dari tim pengembang ke tim penguji secara formal di awal pengujian test.

2. **A Test Log**

Catatan yang menyimpan apa yang terjadi selama pengujian dijalankan.

3. **A Test Incident Report**

Menjelaskan setiap kejadian yg terjadi selama pengujian dijalankan yang membutuhkan investigasi lebih lanjut.

4. **A Test Summary Report**

Ringkasan aktifitas pengujian yang berhubungan dengan satu atau lebih spesifikasi desain pengujian.

4. Facts About Software Engineering:



1. The best programmers are up to 28 times better than the worst programmers.

(Programmer terbaik lebih baik hingga 28 kali daripada programmer terburuk)



2. New tools / techniques cause an initial LOSS of productivity / quality.

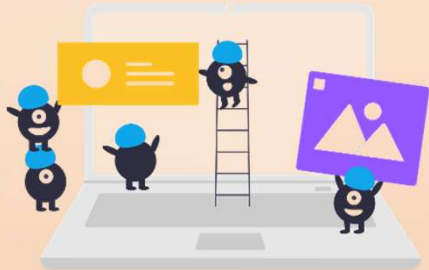
(Alat / teknik yang baru dapat menyebabkan KERUGIAN di awal produktivitas / kualitas)



3. The answer to a feasibility study is almost always “yes”.

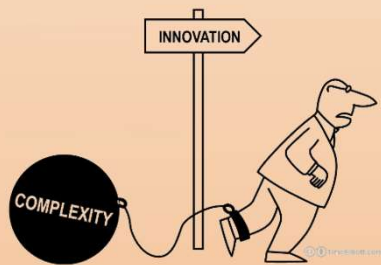
(Jawaban untuk studi kelayakan hampir selalu “ya”)

4. Facts About Software Engineering:



4. Reusable components are three times as hard to build.

(Komponen guna ulang tiga kali lebih sulit untuk dibangun)



5. For every 25% increase in problem complexity, there is a 100% increase in solution complexity.

(Untuk setiap 25% peningkatan kompleksitas masalah, terdapat peningkatan 100% dalam kompleksitas solusinya)



6. 80% of software work is intellectual. A fair amount of it is creative. Little of it is clerical.

(80% pekerjaan perangkat lunak adalah intelektual. Cukup banyak kreatifitas, dan sebagian kecil klerikal)

4. Facts About Software Engineering:



7. Requirements errors are the most expensive to fix during production.

(Kesalahan *requirement* adalah hal paling mahal untuk diperbaiki selama proses produksi)



8. Missing requirements are the hardest requirement errors to correct.

(Kehilangan *requirement* adalah kesalahan *requirement* yang paling sulit untuk diperbaiki)



9. Error-removal is the most time-consuming phase of the life cycle.

(Penghapusan kesalahan adalah fase siklus hidup yang paling banyak memakan waktu)

4. Facts About Software Engineering:



10. Software is usually tested at best at the 55-60% (branch) coverage level.

(Perangkat lunak biasanya diuji paling baik pada tingkat cakupan 55-60% (cabang))



11. Rigorous inspections can remove up to 90% of errors before the first test case is run.

(Inspeksi yang ketat dapat menghilangkan hingga 90% kesalahan sebelum kasus uji pertama dijalankan)



12. Maintenance typically consumes 40-80% of software costs. It is probably the most important life cycle phase of software.

(Pemeliharaan biasanya menghabiskan 40-80% dari biaya perangkat lunak. Hal ini bisa jadi merupakan fase siklus hidup perangkat lunak yang paling penting)

4. Facts About Software Engineering:



13. Enhancements represent roughly 60% of maintenance costs.

(Peningkatan mewakili sekitar 60% dari biaya pemeliharaan)



14. There is no single best approach to software error removal.

(Tidak ada pendekatan tunggal terbaik untuk penghapusan kesalahan perangkat lunak)



End of File