Pengantar IF3267 Pengujian Perangkat Lunak

FATAN KASYIDI

Aturan Kelas

Tidak ada ujian susulan/tambahan seperti Quiz kecuali untuk UTS dan UAS dengan syarat sakit (rs)

Tidak ada protes di akhir kuliah (setelah indeks akhir keluar) kecuali kesalahan rekapitulasi

TIDAK ADA REMEDIAL

Tingkat kehadiran harus memenuhi minimal 80 %, atau hanya boleh tidak hadir maksimal 3 kali, Jika tidak memenuhi, maka nilai akhir otomatis E

Jika diakhir perkuliahan, terdapat mahasiswa mendapat nilai T atau K, wajib untuk melaporkan hal tersebut ke dosen bersangkutan. Apabila selama 1 minggu tidak melapor, maka nilai otomatis berubah menjadi D/E

Kecurangan akademik akan menyebabkan nilai akhir E

- **Menjadi sumber kecurangan juga E (pemberi bantuan utk curang)**
- **Kecurangan tugas berkelompok : E untuk semua anggota**

Keterlambatan masuk kelas maks 20 menit, selebihnya tidak diperkenankan masuk

Berpakaian rapi, sopan dan pantas

Mata Kuliah

IF3267 Pengujian Perangkat Lunak

Semester 6

2 SKS

Dosen Pengampu :

- Fatan Kasyidi, S.Kom., M.T.
- Yulison H. Chrisnanto, S.T., M.T.

Jadwal Kuliah

- Kamis, 15.00, DSE-A, R.1-3
- Kamis, 13.00, DSE-B, R.1-3

Persentase Penilaian

Kehadiran	: 5%
Tugas/Latihan, Quiz	: 15%
Tugas Besar (Presentasi)	: 25% (Tentatif)
UTS	: 25%
UAS	: 30%
Total	: 100%

*Sewaktu-waktu dapat berubah

Silabus

1. Pengantar Kuliah
2. Posisi Pengujian pada SDLC, Srategi Pengujian
3. Strategi Pen
4. Pengujian Unit (Black Box)
5. Pengujian White box
6. Pengujian Black box
7. Pengujian Integrasi
8. Pengujian Sistem
9. User Acceptance Test
10. Dokumentasi
11. Deployment dan Maintenance

Google Classroom

Setiap mahasiswa wajib masuk ke classroom mata kuliah Pengujian Perangkat Lunak.

Kode masuknya :

DSE-A : veih2aj

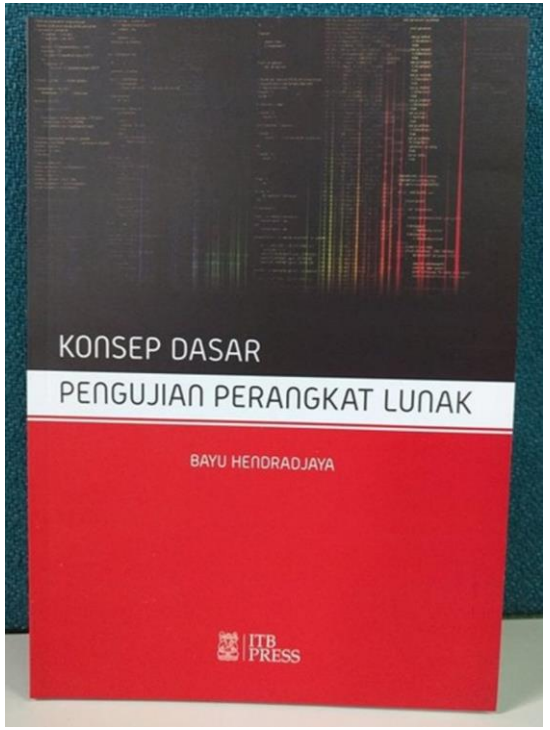
DSE-B : qtcnpbh

Referensi

Sommervile, Ian. 2016. “Software Engineering 10th”. Pearson Education Limited. England

S. Pressman, Roger. 2010. “Software Engineering: A Practitionr’s Appriach 7th”. New York. Amerika.

Hendradjaya, Bayu. 2017. “Konsep Dasar Pengujian Perangkat Lunak”. ITB Press. Bandung. Indonesia



Materi Pendahuluan

Definisi Software Testing

- Myers (1976) : Pengujian adalah proses mengeksekusi dengan tujuan mencari kesalahan (defect/error) [Myers, 1976]
- Hetzel (1988) : pengujian adalah proses mementuk kepercayaan bahwa program atau sistem akan dapat melakukan yang diharapkan [Gelperin and Hetzel, 1988; Hetzel, 1991]
- Dijktra (1972) : pengujian hanya dapat menunjukkan kesalahan, tetapi tidak dapat menunjukkan ketiadaan kesalahan [Dahl ey al., 1972] → pengujian diusahakan selengkap mungkin

Tujuan Software Testing

1. Menunjukkan keberadaan defect software
2. Menunjukkan adanya perbedaan antara requirement spesificatin dengan code implementation
3. Menunjukkan performansi dari software
4. Menunjukkan kualitas dari software

Verification and Validation

Software testing adalah bagian dari proses validation dan verification (V&V)

→ untuk mencari berbagai macam error dan problem

→ dapat mengarahkan developer untuk memperbaiki defect yang muncul

- ✓ **Verification** adalah proses yang merujuk kepada serangkaian tugas yang memastikan bahwa software telah mengimplementasikan fungsi tertentu dengan benar
- ✓ **Validation** adalah proses yang merujuk kepada serangkaian tugas yang memastikan bahwa software telah dibangun sesuai dengan user requirement

Tujuan : memastikan bahwa software yang sedang dibangun telah "fit for purpose"

Verification and Validation (2)

Barry Boem, seorang perintis software engineering menyatakan V&V dengan (Boem, 1979),

Verification : “Are we building product right ?”

Validation : “Are we building the right product ?”

	Validation	Verification
Tujuan	membuktikan apakah produk yang benar sudah dikembangkan	Mengevaluasi apakah pengembangan produk sudah benar
Teknik Pengujian	Pengujian tradisional yang berbasis pada eksekusi program → Program diuji dengan beberapa kasus uji	Inspeksi dan pengkajian ulang (review)

Jenis kesalahan Pengujian

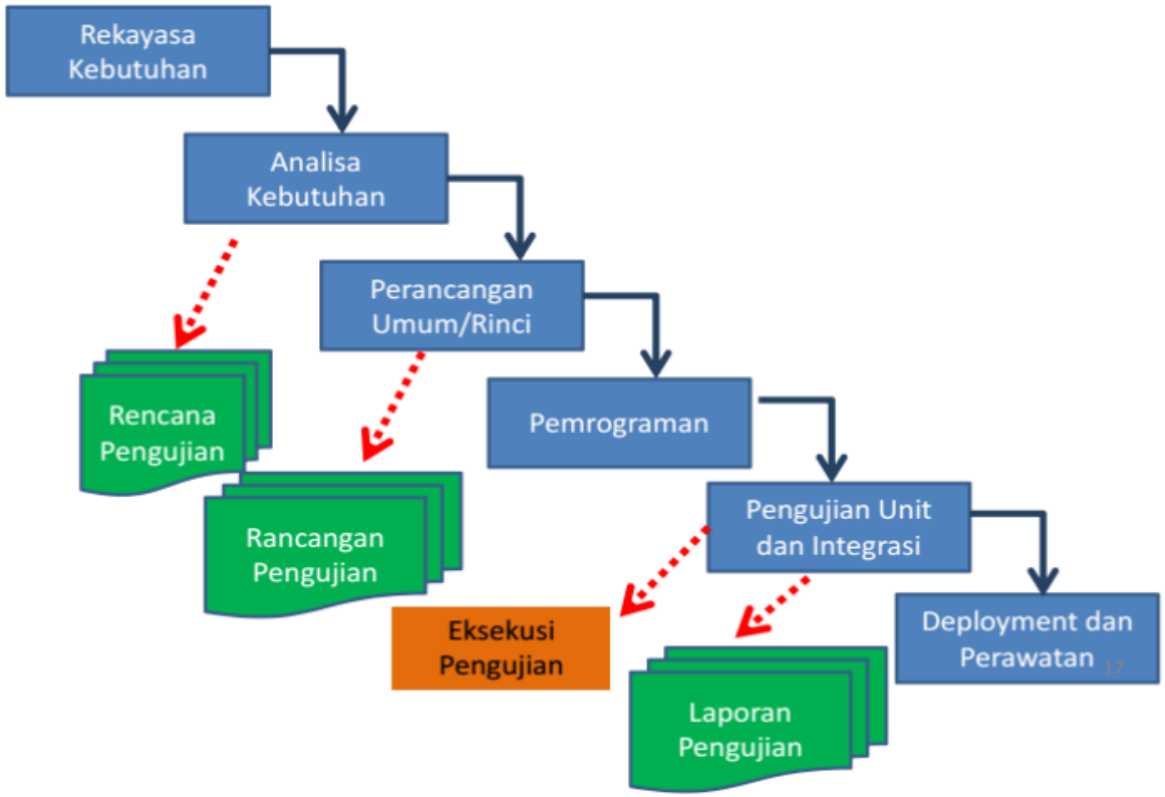
- Software failure, bentuk kegagalan yang terjadi karena ada perilaku dari aplikasi yang tidak sesuai dengan requirement spesification
- Software error, merupakan status dari sistem. Status error yang tidak diperbaiki menyebabkan software failure
- Fault, bug atau defect, mengacu pada sumber terjadinya kesalahan (error)

Contoh menggambarkan hubungannya

> “Suatu program aplikasi perbankan telah dianggap gagal beroperasi (software failure). Kegagalan (failure) disebabkan oleh kesalahan (error) pada bagian perhitungan bunga. Sumber kesahalan (fault/bug/defect) ditemukan pada fungsi kalkulasi CalcInterest()”

Fault/bug/defect → software error → software failure

Aktifitas Pengujian



Perencanaan Uji meliputi:

- Perencanaan apa saja yang akan diuji
- Perencanaan waktu pengujian
- Perencanaan alokasi sumber daya : jumlah penguji (termasuk kemampuan), hardware, software, tools pengujian, lingkungan pengujian, dsb
- Persiapan prosedur uji

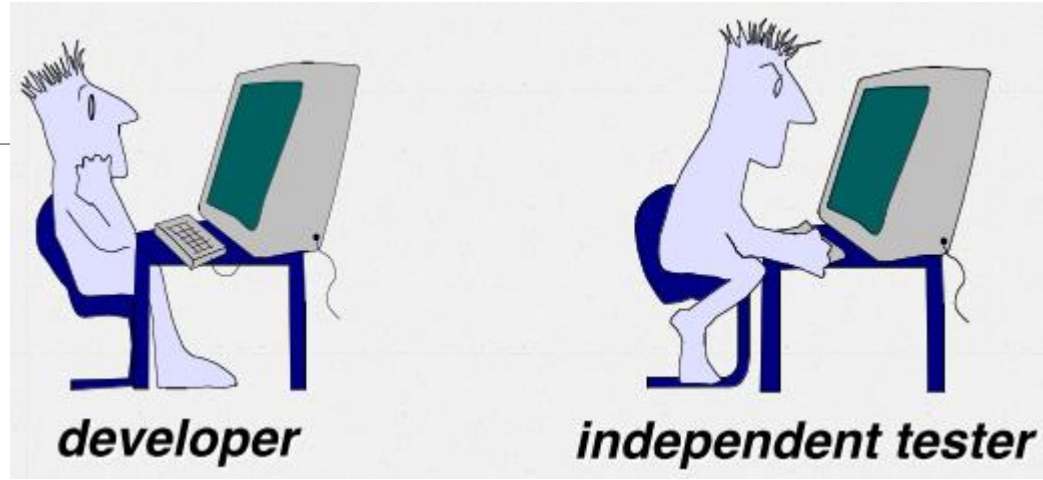
Perancangan Pengujian mencangkup pengembangan kasus uji (test case), yang berisi

- Data masukan (input), data test
- Harapan pengeluaran pengujian (expectation result)
- Cara menilai hasil eksekusi pengujian

Eksekusi pengujian : melakukan pengujian berdasarkan prosedur uji dan kasus uji → hasil lolos dan tidak lolos

Laporan pengujian : rangkuman hasil pengujian untuk kemudian diperbaiki (bug fixing)

Pelaku Pengujian



Developer	Independent Tester
mengetahui system	harus mempelajari system
Melakukan pengujian bahwa program telah memenuji requirement spesification	Melakukan pengujian secara obyektif
Tujuan : membuktikan bahwa program benar	Tujuan : membuktikan program salah
Gagal jika ditemukan kesalahan	Berhasil jika ditemukan kesalahan

Pengujian Efektif

Aktifitas Pengujian adalah kegiatan “Mahal”

Sehingga sedapat mungkin dilakukan secara efektif dengan berbagai teknik yang ada

Efektifitas (keberhasilan usaha agar tercapai tujuan) lebih **diutamakan** dibandingkan **efisiensi** (pencapaian hasil dengan usaha seminimalis mungkin)

Panduan pengujian efektif

1. Lakukan **tecnical review** untuk mengkaji ulang
2. Kelengkapan requirement spesification dan perancangan PL
3. Strategi pengujian dan rancangan uji kasus
4. Pengujian dimulai dari elemen/modul/unit yang paling dasar (**testing in small)** kemudian pengujian integrasi antar unit sehingga semua komponen unit terintegrasi secara lengkap **(testing in large)**
5. Gunakan teknik pegujian dengan pendekatan pengembangan, misalnya berorientasi obyek
6. Pengujian dimulai dari developer kemudian dilanjutkan oleh independent tester
7. Mengerti profile end-user
8. Lain-lain : pengelaman penguji, pengembangan kasus uji yang baik (lakukan pada titik yang sering terjadi error, hindari duplikasi kasus uji, gunakan critical case)

Dokumentasi Pengujian

Standar Dokumentasi pengujian merujuk IEEE 829

1. Test plan, dokumen perencanaan pengujian - ***Plan how the testing will proceed.***
2. Test design spesification, dokumen spesifikasi perancangan pengujian – (***decide what needs to be tested***) pendekatan pengujian baik satu atau kombinasi fitur PL
3. Test case spesification, dokumen spesifikasi kasus pengujian – (***create the tests to be run***) data input, expectation result, kondisi pengujian
4. Test procedure spesification, dokumen spesifikasi prosedur pengujian – (***describe how the tests are run***) sekumpulan aksi dalam mengeksekusi pengujian
5. Test item transmilttal report, dokumen laporan item - ***specify the items released for testing***.
6. Test log, dokumen log pengujian – (***record the details of tests in time order***) catatan kronologis hasil pengujian pada semua kasus
7. Test incidedent report, dokumen laporan insiden pengujian – (***record details of events that need to be investigated***) catatan hasil pengujian yang tidak sesuai spesifikasi untuk ditindaklanjuti
8. Test summary report, dokumen laporan hasil pengujian – (***summarise and evaluate tests***) hasil aktifitas pengujian pada item pengujian

Dokumentasi Pengujian (2)

Lebih Lanjut, dokumentasi pengujian :

https://www.coleyconsulting.co.uk/IEEE829.htm