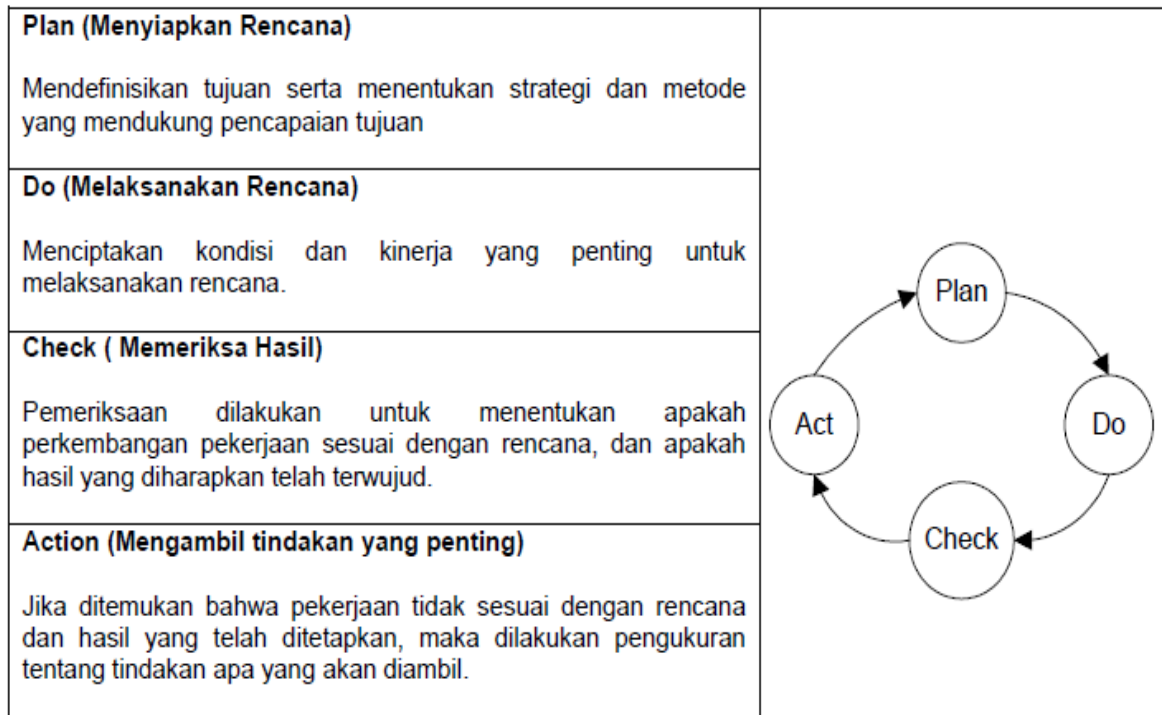


BAB 1

PENDAHULUAN

Secara umum, diketahui bahwa dalam suatu siklus pengembangan perangkat lunak selalu terdapat empat proses utama, yaitu :



Gambar Siklus Pengembangan secara umum

Penamaan untuk empat proses di atas mungkin akan berbeda, dan pentahapan proses dalam siklus pengembangan juga akan berbeda. Keempat proses tersebut dapat tersebar dalam proses yang ada dalam siklus ataupun dapat berada dalam beberapa proses yang berbeda.

Pelaksanaan kegiatan pada tahap analisis, desain dan implementasi di dalam siklus pembuatan perangkat tidak menjamin bahwa suatu perangkat lunak akan bebas dari kesalahan (*fault free*), untuk mengurangi atau bahkan menghilangkan kesalahan pada perangkat lunak diperlukan suatu tahap pengujian. Kesalahan yang terjadi juga tidak hanya kesalahan yang dapat menyebabkan fungsi perangkat lunak tidak dapat berjalan (*error*), tetapi dapat juga berarti penggunaan perangkat lunak sukar untuk dimengerti user dan proses pelacakan kesalahan sukar untuk dilakukan.

Pengujian dapat berarti proses untuk mengecek apakah suatu perangkat lunak yang dihasilkan sudah dapat dijalankan sesuai dengan standar tertentu. Standar yang dijadikan acuan dapat berupa menurut instansi tertentu ataupun disesuaikan dengan keperluan customer/user.

Pengertian pengujian dari masa ke masa :

1. Memantapkan kepercayaan bahwa program melakukan apa yang harus dikerjakan.
2. Proses mengeksekusi suatu program atau sistem dengan tujuan mencari kesalahan.
3. Mendeteksi kesalahan spesifikasi dan penyimpangan dari spesifikasi tersebut.
4. Semua aktivitas yang ditujukan saat evaluasi suatu atribut atau kemampuan program atau sistem.
5. Pengukuran kualitas Perangkat lunak.
6. Proses mengevaluasi suatu program atau sistem.
7. Memverifikasi bahwa suatu sistem memuaskan atau memenuhi requirement tertentu atau mengidentifikasi perbedaan antara yang diharapkan dengan hasil yang ada
8. Memberitahukan bahwa program melakukan suatu fungsi yang diharapkan secara benar (layak).
9. Proses menjalankan dan mengevaluasi sebuah perangkat lunak secara manual maupun otomatis untuk menguji apakah perangkat lunak sudah memenuhi persyaratan atau belum
10. Untuk menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya

Berdasarkan definisi di atas, maka dapat disimpulkan bahwa pengujian dilakukan untuk memenuhi persyaratan kualitas perangkat lunak, dengan cara mengeksekusi program untuk mencari kesalahan sintaks program, melakukan verifikasi perangkat lunak untuk melihat kesesuaian antara perangkat lunak dengan keinginan customer.

Definisi Pengujian menurut IEEE dan ANSI :

- The process of operating a system or component under specified condition, observing or recording the result, and making an evaluation of some aspect of system/component. (IEEE/ANSI, 1990 std 610.12-1990)
- The process of analyzing software item to detect the difference existing and required condition (that is, bugs) and to evaluate the feature of the software items. (IEEE/ANSI, 1983 std 829-1983)

(IEEE – Institute of Electrical and Electronics Engineering, ANSI – American National Standards Institute)

Berdasarkan kedua definisi di atas dapat disimpulkan bahwa pengujian perangkat lunak adalah proses untuk mencari kesalahan pada setiap item perangkat lunak, mencatat hasilnya, mengevaluasi setiap aspek pada setiap komponen system dan mengevaluasi semua fasilitas dari perangkat lunak yang dikembangkan.

Terdapat 2 hal utama yang dilakukan dalam pengujian, yaitu :

- Verifikasi adalah proses mengevaluasi suatu system/component untuk menentukan apakah suatu produk yang diselesaikan setelah fase pengembangan memenuhi kondisi seperti yang telah ditetapkan pada awal pengembangan (saat menentukan spesifikasi) perangkat lunak.
(“Are we building the product right?”)
- Validasi adalah proses mengevaluasi suatu system/komponen pada akhir atau selama masa pengembangan untuk menentukan apakah produk yang dihasilkan telah memenuhi kebutuhankebutuhan dan persyaratan tertentu yang diminta oleh user.
(“Are we building the right product?”).

Definisi kualitas

- Menurut Crosby
Kualitas adalah pemenuhan terhadap kebutuhan
- Menurut ISO-8402
Kualitas adalah keseluruhan dari fitur yang menjadikan suatu produk dapat memuaskan atau dipakai sesuai kebutuhan dengan harga yang terjangkau
- Menurut W.E Perry
Kualitas adalah pemenuhan terhadap standar

- Menurut R.Glass
Kualitas adalah tingkat kesempurnaan
- Menurut J.Juran
Kualitas adalah tepat guna

Hubungan testing dan kualitas

Software yang berkualitas adalah software yang bebas error dan bug secara objektif, tepat waktu dan dana, sesuai dengan kebutuhan atau keinginan dan dapat dirawat (maintainable)

Definisi objektif :

Suatu proses pembuktian yang terstruktur, terencana dan terdokumentasi dengan baik

- Testing membuat kualitas dapat dilihat secara objektif, karena testing merupakan pengukuran dari kualitas software
- Testing tidak dapat memastikan kualitas software, namun dapat memberikan jaminan terhadap software pada suatu tingkat tertentu
- Jaminan kualitas (Quality Assurance – QA) mengukur kualitas proses yang digunakan untuk membuat produk berkualitas
Testing merupakan bagian dari aktifitas QA
- Proyek pengembangan software memiliki kecenderungan untuk mengalami kegagalan

Salah satu usaha menurunkan tingkat resiko terjadinya kegagalan adalah dengan berorientasi pada kualitas

Tujuan testing :

Adalah menemukan sebanyak mungkin masalah (error)

Tujuan dari menemukan masalah adalah memperbaikinya. Tangani masalah yang bersifat penting, karena tidak semua permasalahan dapat diselesaikan (berlaku prinsip pareto)

Misi tim testing

Tidak hanya untuk melakukan testing, tetapi juga membantu meminimalkan resiko kegagalan proyek

- Mencari masalah dari produk
- Mencari masalah potensial
- Mencari kehadiran masalah

Intinya adalah mencari dan melaporkan sehingga tim lain dapat membuat keputusan terhadap pengembangan produk

Perlu diingat, tester hanya menginformasikan Tidak melakukan pembenahan kode. Tester adalah individu yang memberikan hasil pengukuran dari kualitas produk

Psikologi testing

Jika pengembangan dilakukan secara konstruktif, maka testing dilakukan secara destruktif. Tester harus mempunyai keinginan yang mendasar untuk membuktikan kode gagal dan akan melakukan apa saja untuk membuatnya gagal. Bila seorang tester hanya ingin membuktikan bahwa kode beraksi sesuai dengan fungsi bisnisnya, maka tester tersebut telah gagal dalam menjalankan tugasnya sebagai tester

Prinsip-prinsip testing

1. Testing yang komplit (menyeluruh) tidak memungkinkan dilakukan karena :

- Kemungkinan jumlah kombinasi testcase yang amat besar
 - Pertimbangan domain masukan yang mungkin sangat besar jumlahnya (masukan yang valid, tdk valid, masukan yg diedit dll)
 - Kompleksitas user interface dan desain
 - Jalur program yang mungkin dapat dilewati sangat banyak
- Harus dilakukan test ulang, setiap ada perbaikan pada masing-masing bug

2. Testing merupakan pekerjaan yang kreatif dan sulit

- Mitos-mitos yang salah tentang testing :
 - Testing itu mudah
 - Tiap orang akan dapat melakukan testing dengan sendirinya

- Padahal testing bukanlah hal yang sederhana, karena
 - Untuk dapat melakukan testing yang efektif harus mengetahui keseluruhan sistem
 - Sistem sendiri tidak sederhana (tidak mudah dipahami)

3. Testing berbasis pada resiko

Walaupun testing secara keseluruhan tidak dapat dilakukan *tidak berarti* bahwa testing yang efektif tidak dapat dilakukan

Testing merupakan hasil pertimbangan dari resiko dan ekonomi

Secara ringkas, testing dipengaruhi oleh pertimbangan :

- Sumber daya dan biaya yang dibutuhkan untuk melakukan testing menurut skala prioritas, kompleksitas dan kesulitan testing
- Biaya dari keterlambatan pengiriman produk (kemungkinan besar disebabkan testing)
- Kemungkinan adanya suatu defect
- Biaya yang disebabkan oleh defect, bilamana defect tsb menyebabkan error yang membawa kerugian langsung maupun tak langsung bagi customer

4. Testing harus direncanakan

Butuh pemikiran dgn pendekatan secara keseluruhan, desain tes dan penetapan hasil yg diinginkan utk setiap kasus tes (test case) yg dipilih

- test plan : dokumen yg mencakup keseluruh tujuan testing dan pendekatan testing
- test design: dokumen yg mendefinisikan apa yg telah dipilih utk dites dan hasil yg diharapkan

“test direncanakan dan didesain sebelum kode dibuat”

Perencanaan tes sangat penting, yaitu :

- Utk dpt menjaga arah pelaksanaan tes agar tidak menyimpang dari tujuan tes itu sendiri (mengukur kualitas SW)
- Menjaga kesesuaian penggunaan sumber daya dan jadual proyek dengan menetapkan apa yang akan dites dan kapan berhenti
- Membantu tester fokus terhadap apa yang akan dites (membuat test case)

5. Testing butuh independensi

Testing yang paling efektif adalah yang dilakukan oleh pihak ketiga (tidak bias)

Dasar-dasar testing

- Testing dikatakan baik, jika :
 - Kemungkinan mendapatkan error tinggi
 - Tidak redundan --> resource terbatas, tiap tes yang dilakukan HARUS memiliki tujuan yang berbeda
- Tidak terlalu simpel atau kompleks
 - Kode untuk kondisi khusus mendapatkan porsi tes yang sama dengan kode yang umum
 - Kode untuk kondisi khusus --> peluang memp. bug tinggi
 - Testing untuk kode yang dijalankan dalam kondisi normal tidak mengeksekusi error handling code
 - Testing yang sukses adalah testing yang berhasil menemukan error yang tidak diketahui