



**UNIVERSITAS
BINA INSANI**

TESTING DAN IMPLEMENTASI SISTEM

PRINSIP-PRINSIP TESTING

Ahmad Chusyairi, M.Kom



binainsani.ac.id



Bina Insani University



[binainsaniuniversity](https://www.instagram.com/binainsaniuniversity)



[@BinaInsaniOK](https://twitter.com/BinaInsaniOK)

Jl. Raya Siliwangi No. 6 Rawa Parjang - Bekasi

Rencana Kegiatan Pembelajaran Mingguan

Capaian Pembelajaran Pertemuan

Mahasiswa mengetahui dan memahami prinsip-prinsip *testing*

Kemampuan Akhir Capaian Pembelajaran

Mahasiswa mampu menjelaskan dan memahami prinsip-prinsip *testing*

Bahan Kajian

Prinsip-Prinsip *Testing*:

- 🐼 Obyektifitas *Testing*
- 🐼 Misi dari Tim *Testing*
- 🐼 Prinsip-Prinsip *Testing*

Penilaian

Mahasiswa mampu menjelaskan obyektifitas *testing*, misi dari tim *testing*, prinsip-prinsip *testing*

- 👤 Secara umum **obyektifitas** dari ***testing*** adalah melakukan verifikasi, validasi dan deteksi *error* untuk menemukan masalah dan tujuan dari penemuan ini adalah membenahinya.
- 👤 Bagian dari obyektifitas *testing*, antara lain:
 - 🔥 Meningkatkan kepercayaan bahwa sistem dapat digunakan dengan tingkat resiko yang dapat diterima.
 - 🔥 Menyediakan informasi yang dapat mencegah terulangnya *error* yang pernah terjadi.
 - 🔥 Menyediakan informasi yang membantu untuk deteksi *error* secara dini.
 - 🔥 Mencari sejauh apa kemampuan dari sistem.
 - 🔥 Menyediakan informasi untuk kualitas dari produk *software*.

Misi dari Tim *Testing*

- 🦊 **Misi dari tim *testing*** tidak hanya untuk melakukan *testing*, tapi juga untuk membantu meminimalkan resiko kegagalan proyek.

- ❗ **Tester** adalah individu yang memberikan hasil pengukuran dari kualitas produk.
- ❗ *Tester* tidak melakukan pembenahan atau pembedahan kode, tidak memermalukan atau melakukan komplain pada suatu individu atau tim, hanya menginformasikan.
- ❗ *Tester* mencari manifestasi masalah dari produk, masalah yang potensial, dan kehadiran dari masalah.
- ❗ *Tester* mengeksplorasi, mengevaluasi, melacak, dan melaporkan kualitas produk, sehingga tim lainnya dari proyek dapat membuat keputusan terhadap pengembangan produk.

- 🦄 **Testing** merupakan suatu psikologi yang menarik, jika pengembangan dilakukan secara konstruktif, maka testing adalah *destruktif*.
- 🦄 Seorang *developer* bertugas membangun, sedangkan seorang *tester* justru berusaha untuk menghancurkan. Mental yang seperti inilah yang penting bagi seorang *tester*.
- 🦄 Jika seorang *desainer* menanamkan testabilitas, programmer berorientasi pada *zero defect minded*, maka *tester* mempunyai untuk pembuktian kode gagal.
- 🦄 *Tester* ingin membuktikan bahwa kode beraksi sesuai dengan fungsi bisnisnya, maka *tester* tersebut gagal dalam menjalankan tugasnya.

Prinsip-Prinsip *Testing*

Terdapat **6 kunci prinsip-prinsip *testing***, yaitu:

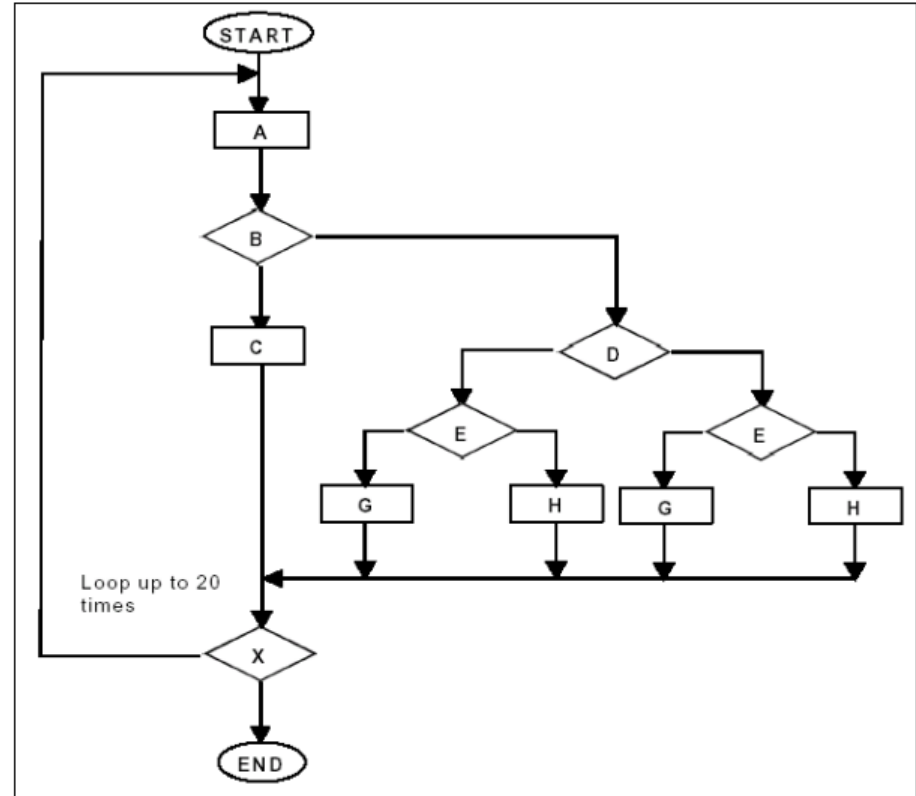
- 🦿 *Testing* yang komplit tidak mungkin
- 🦿 *Testing* merupakan pekerjaan yang kreatif dan sulit.
- 🦿 Alasan yang penting diadakannya testing adalah untuk mencegah terjadinya error
- 🦿 *Testing* berbasis pada resiko
- 🦿 *Testing* harus direncanakan
- 🦿 *Testing* membutuhkan independensi

Testing yang Komplit Tidak Mungkin

- 👤 **Testing yang komplit** secara menyeluruh **tidak mungkin** dilakukan, karena jumlah kemungkinan kombinasi *test case* yang besar dengan pertimbangan:
 - 🔥 Domain masukan, di mana harus dilakukan tes semua masukan yang valid, semua kemungkinan yang tidak valid, semua masukan yang diedit, semua variasi masukan berdasarkan pada waktu kejadian, oleh karena itu dibutuhkan prioritas dalam pemilihan domain masukan dari sistem yang akan dites.
 - 🔥 Kompleksitas, *user interface* dan desain sangat kompleks yang dilakukan *testing* secara komplit. Jika suatu kesalahan desain terjadi, bagaimana tester dapat menyatakan *bug* di mana ada dalam spesifikasi, dan mengenali *bug* dalam sistem.

Testing yang Komplit Tidak Mungkin (2)

🔥 Jalur Program, akan terdapat banyak jalur yang dilewati pada suatu program yan dites secara komplit.







Testing yang Komplit Tidak Mungkin (2)

- ❧ *Testing* tidak untuk membuktikan kebenaran sistem namun keberadaan *error* dengan kondisi-kondisi yang mempunyai kesamaan secara mendasar dengan yang diteskan.
- ❧ Jumlah *error* pada sistem dapat diprediksi dengan akurasi tertentu, namun tidak dapat menjamin akan tidak adanya *error* lain pada sistem.
- ❧ ***Testing* secara menyeluruh adalah tidak mungkin**, karena perhitungan yang terjadi pada program sangat besar dan kompleks. Jadi tidak mungkin melakukan *testing* pada tiap kemungkinan kombinasi perhitungan secara menyeluruh dan yang mungkin adalah melakukan *testing* logika dari program dan semua kondisi dari semua level komponen telah diperiksa.

Testing merupakan Pekerjaan yang Kreatif dan Sulit

Testing merupakan pekerjaan yang sulit dan untuk dapat sukses dalam melakukan *testing* dibutuhkan:


-  Kreatifitas
-  Pengetahuan Bisnis
-  Pengalaman *Testing*
-  Metodologi *Testing*

Alasan yang Penting diadakannya *Testing* adalah untuk Mencegah Terjadinya *Error*

Konsep siklus dari testing:

 *Testing* bukan untuk satu fase pengembangan saja

 Hasil *testing* diasosiasikan pada tiap fase pengembangan

 Semua *testing* harus dapat dilacak dan memenuhi kebutuhan dari konsumen dan salah satu obyektifitas dari *testing* adalah memperbaiki *error*.

Testing Berbasis pada Resiko

Testing merupakan hasil pertimbangan dari resiko dan ekonomi dengan 4 faktor utama:

- 🦿 Sumber daya dan biaya yang dibutuhkan untuk melakukan *testing* berdasarkan pada skala prioritas, kompleksitas dan kesulitan *testing*.
- 🦿 Biaya dari keterlambatan pengiriman produk
- 🦿 Kemungkinan adanya suatu *defect*
- 🦿 Biaya yang disebabkan oleh *defect*

Testing harus direncanakan

- ❧ Suatu dokumen yang mencakup keseluruhan dari tujuan testing dan pendekatan testing disebut Rencana Tes (***Plan Test***).
- ❧ Suatu dokumen atau pernyataan yang mendefinisikan apa yang telah dipilih untuk dites dan menjelaskan hasil yang diharapkan disebut Desain Tes (***Test Design***).

Rencana Tes	Desain Tes
❧ Pernyataan obyektifitas <i>testing</i>	❧ Spesifikasi tes yang dikembangkan
❧ Deskripsi pendekatan tes	❧ Deskripsi pengelompokan tes
❧ Sekelompok tugas untuk mencapai obyektifitas <i>testing</i>	

Testing harus direncanakan (2)

- ❧ Apa yang menjadi penilaian suatu tes tertentu benar?
 - 🔥 Kepercayaan akan apa yang dapat dihasilkan dengan biaya yang digunakan untuk *testing*.
 - 🔥 Adanya penemuan masalah dan *defect*.
- ❧ Perencanaan tes sangat penting, karena:
 - 🔥 Untuk dapat menjaga arah pelaksanaan tes agar tidak menyimpang dari tujuan tes itu sendiri, yaitu untuk mengukur kualitas software.
 - 🔥 Untuk menjaga kesesuaian penggunaan sumber daya dan jadwal proyek
 - 🔥 Untuk membuat *test case* yang baik (tepat guna)

Testing membutuhkan independensi

- 👤 Apa yang disebut *Tester* yang independen (tidak tergantung atau bebas):
 - 🔥 Pengamat yang tidak bias.
 - 🔥 Orang yang bertujuan untuk mengukur kualitas software secara akurat
- 👤 Kunci yang mempengaruhi kinerja dari *testing* adalah:
 - 🔥 Wawasan dan kreatifitas tiap individu yang terlibat.
 - 🔥 Pengetahuan dan pemahaman terhadap aplikasi yang dites.
 - 🔥 Pengalaman *testing*.
 - 🔥 Metodologi testing yang digunakan.
 - 🔥 Usaha dan sumber daya yang dipakai.

Hal ini disebabkan oleh:

- 🐞 Sistem pengembangan yang kurang baik dan terencana.
- 🐞 Sistem pelayanan yang kurang baik dan terencana.
- 🐞 Analis, desainer dan programmer tidak mengetahui bagaimana membangun suatu kualitas ke dalam sistem yang ada.
- 🐞 *Tester* tidak banyak terpengaruh oleh definisi dari kebutuhan, *development* atau proses pelayanan.

Faktor Pendukung Pengujian

🔥 Koordinasi tim baik	🔥 Diberikan akses full terhadap perangkat lunak
🔥 Kompatibel dengan hardware	🔥 Waktu pengujian yang terjadwal
🔥 Perangkat lunak sederhana	🔥 Aplikasi <i>open source</i>
🔥 Tidak dibutuhkan koneksi internet	🔥 Instalasi program mudah dipahami
🔥 Sistem responsif	🔥 Aplikasi berbasis web yang familiar
🔥 Aplikasi belum menggunakan <i>framework</i>	🔥 Aplikasi dilengkapi dengan modul
🔥 Sistem mudah dipahami	🔥 Perangkat lunak mudah digunakan

Faktor Penghambat Pengujian

🔥 File terlalu besar	🔥 Pihak pengembangan kurang kooperatif
🔥 Terjadi pergantian perangkat lunak	🔥 Koneksi internet tidak lancar
🔥 Alur program rumit	🔥 <i>User Interface</i> tidak efisien
🔥 Kurang komunikasi	🔥 Terlalu banyak <i>bug</i>
🔥 Keterlambatan distribusi perangkat lunak	🔥 Perangkat lunak di <i>hack</i>
🔥 Perangkat lunak tidak kompatibel OS	🔥 Perangkat lunak sulit dipahami
🔥 Tidak diberikan catatan untuk login	🔥 Tidak ada dokumentasi sebelumnya
🔥 Penggunaan <i>framework</i> yang asing	

Romeo. 2003. Testing dan Implementasi Sistem Edisi Pertama. STIKOM Surabaya.


Subhiyakto E. R. & Utomo D. W. 2016. Strategi, Teknik, Fakto Pendukung dan Penghambat Pengujian untuk Pengembang Perangkat Lunak Pemula. Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA), pp.236-241.

Terima Kasih

Bigger
Better
Higher



ahmad chusyairi




Ahmad Chusyairi
83 subscribers


[SESUAIKAN CHANNEL](#)
[KELOLA VIDEO](#)

[BERANDA](#)
[VIDEO](#)
[PLAYLIST](#)
[CHANNEL](#)
[DISKUSI](#)
[TENTANG](#)


Upload ▶ PUTAR SEMUA




Lanjutan Studi Kasus Web Sederhana #WebSederhana
26 x ditonton • 4 minggu yang lalu




Implementasi Pemrograman PHP dengan Database SQL
57 x ditonton • 1 bulan yang lalu



Studi Kasus Web Sederhana #Berita
21 x ditonton • 1 bulan yang lalu



Implementasi Store Procedure dan Role dalam...
46 x ditonton • 1 bulan yang lalu



Penggunaan Koneksi MySQL pada Form #KoneksiForm...
33 x ditonton • 1 bulan yang lalu