

# PRAKTIKUM 02: PENGANTAR SYSTEM/SOFTWARE TESTING

by Nurul Firdaus, S.Kom., M.Inf.

## TUJUAN PEMBELAJARAN

Mahasiswa memahami dasar-dasar pengujian sistem/software



## REFERENSI

Roger S. Pressman., 2000. Software engineering: a practitioner's approach. *5th ed : McGraw-Hill series in computer science*

Abran, A., Moore, J.W., Bourque, P., Dupuis, R. and Tripp, L., 2004. Software engineering body of knowledge. *IEEE Computer Society, Angela Burgess.*

"Kualitas Perangkat Lunak Model ISO 9126".

<http://fxekobudi.net/ilmukomputer/kualitasperangkat-lunak-model-iso-9126/>

"BAB 8 Jaminan Kualitas Perangkat Lunak Software Quality Assurance [SQA]".

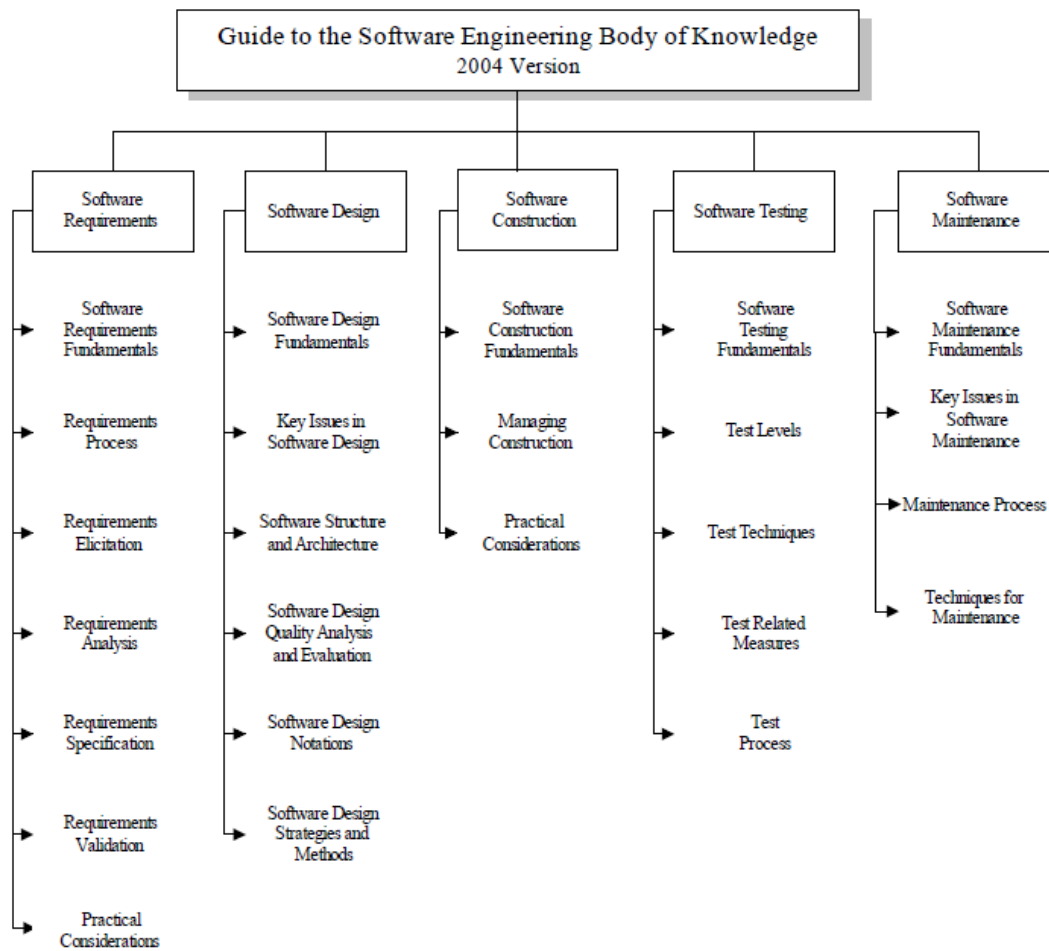
<http://wsilfi.staff.gunadarma.ac.id/Downloads/files/24049/Jaminan+Kualitas+PL.pdf>

"Panduan Dokumen User Acceptance Test (UAT)". <http://dac.telkomuniversity.ac.id/wpcontent/uploads/2015/06/PAKA06A-Panduan-User-Acceptance-Test-UAT-20170410.pdf>

"AlphaTesting".[https://www.tutorialspoint.com/software\\_testing\\_dictionary/alpha\\_testing.htm](https://www.tutorialspoint.com/software_testing_dictionary/alpha_testing.htm)

"BetaTesting".[https://www.tutorialspoint.com/software\\_testing\\_dictionary/beta\\_testing.htm](https://www.tutorialspoint.com/software_testing_dictionary/beta_testing.htm)

# MATERI



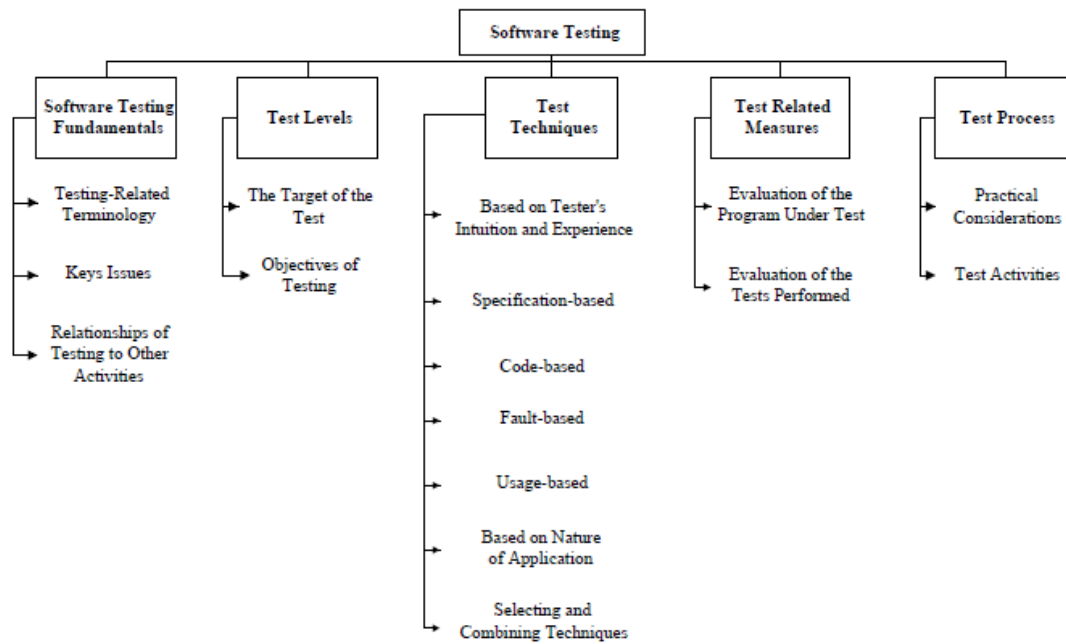


Figure 1 Breakdown of topics for the Software Testing KA.

## Dasar-dasar Pengujian Sistem

a. *System Testing* : Pengujian terhadap integrasi sub-system, yaitu keterhubungan antar *sub-system*.

b. *Acceptance Testing*

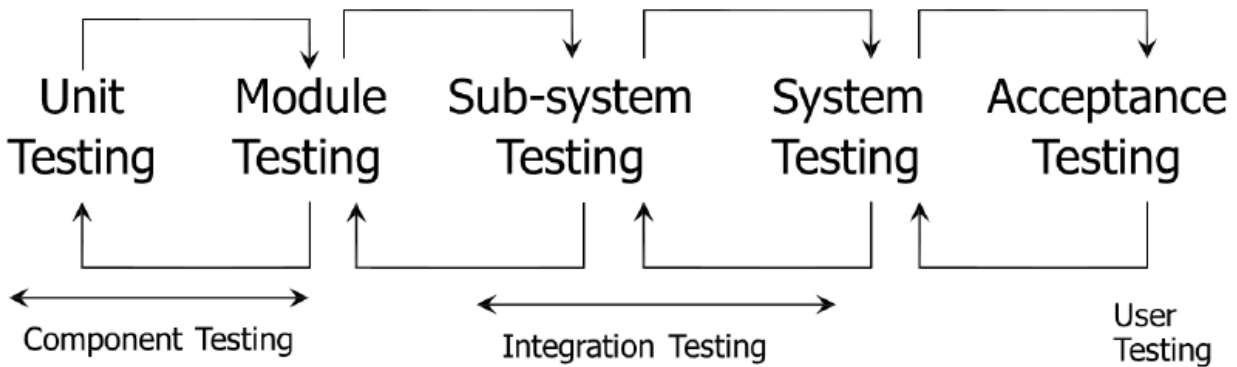
- Pengujian terakhir sebelum sistem dipakai oleh user.
- Melibatkan pengujian dengan data dari pengguna sistem.
- Biasa dikenal sebagai “*alpha test*” (“*beta test*” untuk software komersial, dimana pengujian dilakukan oleh potensial *customer*).

c. *Component Testing*

- Pengujian komponen-komponen program.
- Biasanya dilakukan oleh *component developer* (kecuali untuk sistem kritis)

d. *Integration Testing*

- Pengujian kelompok komponen-komponen yang terintegrasi untuk membentuk *sub-system* ataupun *system*
- Dilakukan oleh tim penguji yang independent
- Pengujian berdasarkan spesifikasi sistem



### Rencana Pengujian

- Proses testing: deskripsi fase-fase utama dalam pengujian.
- Pelacakan kebutuhan: semua kebutuhan user diuji secara individu.
- Item yang diuji: menspesifikasi komponen sistem yang diuji.
- Jadwal testing.
- Prosedur pencatatan hasil dan prosedur.
- Kebutuhan akan *hardware* dan *software*.
- Kendala-kendala, ex: kekurangan staff, alat, waktu dll.

### Test Data dan Kasus Test

- a. Test Data: input yang direncanakan digunakan oleh sistem.
- b. Test Cases: input yang digunakan untuk menguji sistem dan memprediksi output dari input jika sistem beroperasi sesuai dengan spesifikasi.

### Jaminan Mutu Sistem

Jaminan kualitas perangkat lunak (Software Quality Assurance/SQA) adalah aktivitas pelindung yang diaplikasikan pada seluruh proses perangkat lunak. SQA meliputi :

1. Pendekatan manajemen kualitas.
2. Teknologi rekayasa perangkat lunak yang efektif (metode dan pirantinya).
3. Kajian teknik formal yang diaplikasikan pada keseluruhan proses perangkat lunak.
4. Strategi pengujian multitiered (deret bertingkat).
5. Kontrol dokumentasi perangkat lunak dan perubahan yang dibuat untuknya.
6. Prosedur untuk menjamin kesesuaian dengan standar pengembangan perangkat lunak (bila dapat diaplikasikan).
7. Mekanisme pengukuran dan pelaporan.

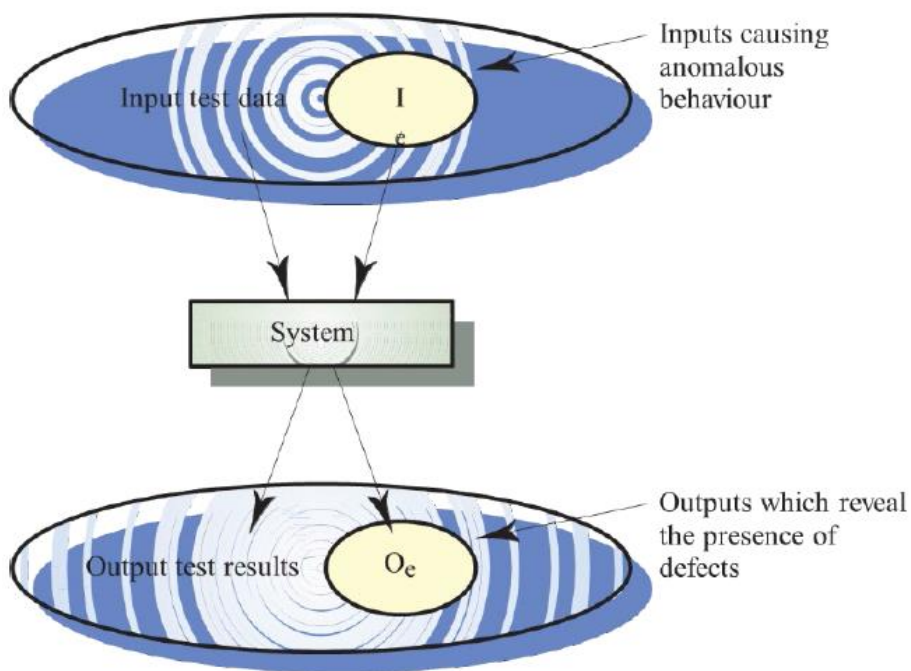
Kualitas perangkat lunak didefinisikan sebagai : Konformansi terhadap kebutuhan fungsional dan kinerja yang dinyatakan secara eksplisit, standar perkembangan yang

didokumentasikan secara eksplisit, dan karakteristik implisit yang diharapkan bagi semua perangkat lunak yang dikembangkan secara profesional. Definisi tersebut berfungsi untuk menekankan tiga hal penting, yaitu:

8. Kebutuhan perangkat lunak merupakan fondasi yang melaluinya kualitas diukur.
9. Standar yang telah ditentukan menetapkan serangkaian kriteria pengembangan yang menuntun cara perangkat lunak direkayasa.
10. Ada serangkaian kebutuhan implisit yang sering tidak dicantumkan.

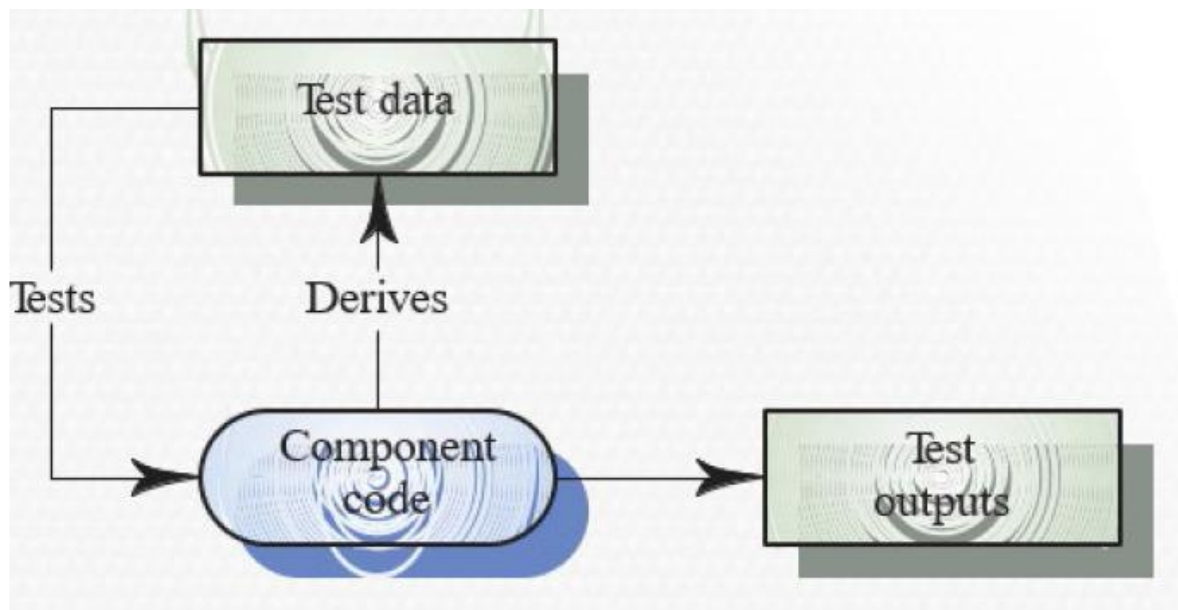
### Black Box Testing

Pendekatan pengujian dimana program dianggap sebagai suatu '*black-box*' ('kotak hitam'). Program *test case* berbasiskan spesifikasi. *Test planning* dapat dimulai sejak awal proses pengembangan system.



### White Box Testing

Sering disebut *structural testing*. Penentuan *test case* disesuaikan dengan struktur sistem. Knowledge program digunakan untuk mengidentifikasi *test case* tambahan. Tujuannya untuk menguji semua statemen program (debug).



### UAT (User Acceptance Test)

User Acceptance Testing (UAT) merupakan proses verifikasi bahwa solusi yang dibuat dalam sistem sudah sesuai untuk pengguna. Proses ini berbeda dengan pengujian sistem (memastikan software tidak crash dan sesuai dengan dokumen permintaan pengguna), melainkan memastikan bahwa solusi dalam sistem tersebut akan bekerja untuk pengguna (yaitu, tes bahwa pengguna menerima solusi di dalam sistem).

UAT umumnya dilakukan oleh klien atau pengguna akhir, biasanya tidak fokus pada identifikasi masalah sederhana seperti kesalahan ejaan, maupun di cacat *showstopper*, seperti crash perangkat lunak. Penguji dan pengembang mengidentifikasi dan memperbaiki masalah ini selama tahap awal pengujian fungsionalitas, pengujian saat integrasi dan pada tahap sistem testing.

Jenis UAT terdiri dari:

- Alpha & Beta Testing
- Contract Acceptance Testing
- Regulation Acceptance Testing
- Operational Acceptance Testing
- Black Box Testing

UAT untuk berorientasi objek diambil dari use case diagram (untuk generalisasi: cukup menerangkan child/anak/spesifikasi nya bukan parent/induk/general nya) sedangkan untuk terstruktur diambil dari DFD level terkecil.

UAT menggunakan data yang berkesinambungan antara use case (fungsi/proses) yang satu dengan yang lain dari awal hingga akhir.

## Alpha and Beta Testing

Fase Pengujian Alfa dan Beta terutama berfokus pada penemuan produk dan produk yang telah mereka berikan gambaran yang jelas tentang bagaimana produk tersebut benar-benar digunakan oleh pengguna waktu nyata. Mereka juga membantu dalam pengalaman pelatihan dengan produk sebelum diluncurkan dan umpan balik yang berharga diterapkan secara efektif untuk meningkatkan kegunaan produk.

Tujuan dan metode Pengujian Alpha & Beta melakukan peralihan di antara mereka sendiri berdasarkan proses yang diikuti oleh proyek dan dapat disesuaikan agar sejalan dengan proses. Kedua teknik pengujian ini telah menghemat ribuan dolar untuk rilis perangkat lunak berskala besar untuk perusahaan seperti Apple, Google, Microsoft, dll.

### Alpha Testing

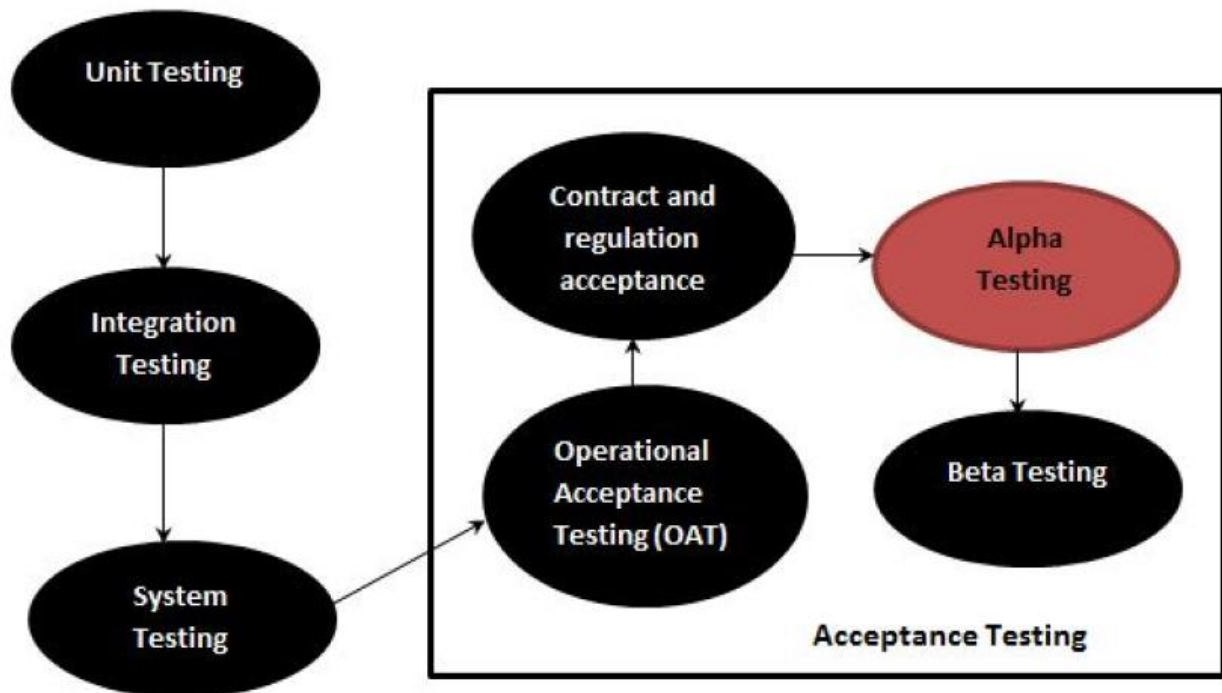
Pengujian alfa adalah salah satu strategi pengujian perangkat lunak yang paling umum digunakan dalam pengembangan perangkat lunak. Terutama digunakan oleh organisasi pengembangan produk.

Tes ini berlangsung di situs pengembang. Pengembang mengamati pengguna dan mencatat masalah. Pengujian alfa adalah pengujian aplikasi saat pengembangan akan selesai. Perubahan desain kecil masih dapat dilakukan sebagai hasil dari pengujian alfa. Pengujian alfa biasanya dilakukan oleh grup yang tidak bergantung pada tim desain, tetapi masih dalam perusahaan, mis. insinyur pengujian perangkat lunak in-house, atau insinyur QA perangkat lunak. Pengujian alfa adalah pengujian akhir sebelum perangkat lunak dirilis ke masyarakat umum. Ini memiliki dua fase:

1. Pada tahap pertama pengujian alfa, perangkat lunak diuji oleh pengembang in-house. Mereka menggunakan perangkat lunak debugger, atau debugger yang dibantu perangkat keras. Tujuannya adalah untuk menangkap bug dengan cepat.
2. Pada tahap kedua pengujian alfa, perangkat lunak diserahkan kepada staf QA perangkat lunak, untuk pengujian tambahan dalam lingkungan yang mirip dengan penggunaan yang dimaksudkan.

Pengujian alfa adalah simulasi atau pengujian operasional aktual oleh calon pengguna / pelanggan atau tim uji independen di situs pengembang. Pengujian alfa sering digunakan untuk perangkat lunak off-the-shelf sebagai bentuk pengujian penerimaan internal, sebelum perangkat lunak digunakan untuk pengujian beta.

Diagram berikut menjelaskan fitment pengujian Alpha dalam siklus hidup pengembangan perangkat lunak.



Pada tahap pertama pengujian alfa, perangkat lunak diuji oleh pengembang internal yang tujuannya adalah untuk menangkap bug dengan cepat.

Pada tahap kedua pengujian alfa, perangkat lunak diberikan kepada tim QA perangkat lunak untuk pengujian tambahan.

Pengujian alfa sering dilakukan untuk perangkat lunak Off-the-shelf komersial (COTS) sebagai bentuk pengujian penerimaan internal, sebelum pengujian beta dilakukan.

### **Beta Testing**

Pengujian Beta juga dikenal sebagai pengujian lapangan. Ini terjadi di situs pelanggan. Ini mengirimkan sistem / perangkat lunak kepada pengguna yang menginstalnya dan menggunakannya di bawah kondisi kerja dunia nyata.

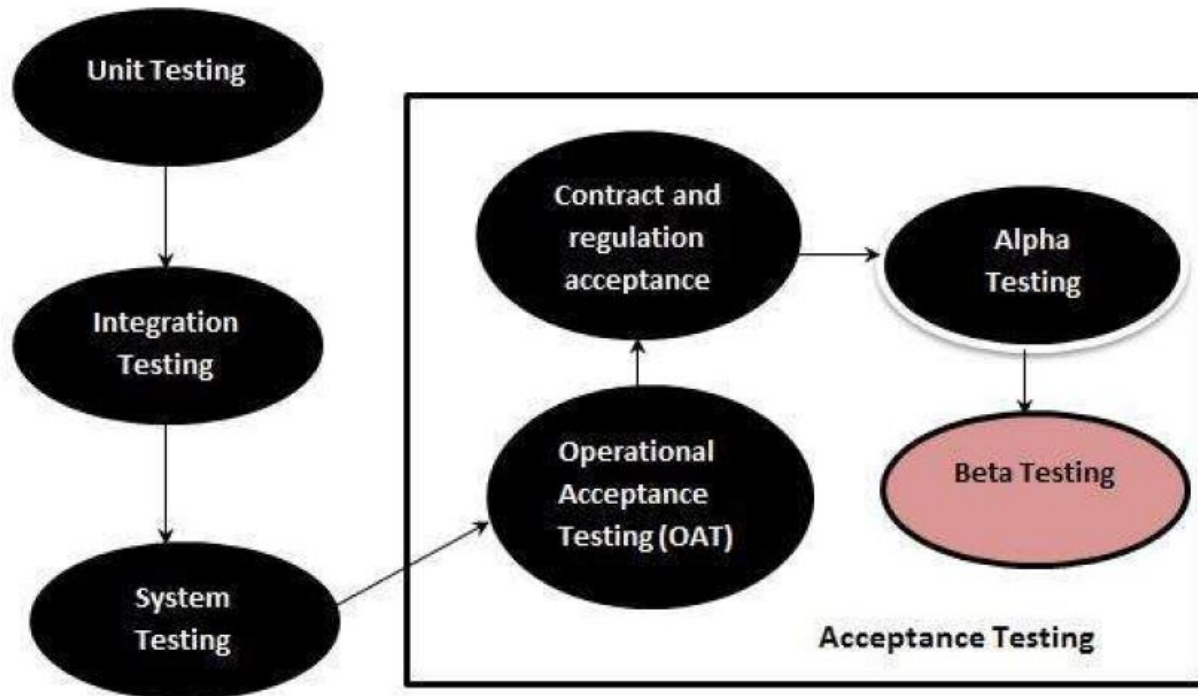
Tes beta adalah tahap kedua pengujian perangkat lunak di mana sampling dari audiens yang dituju mencoba produk tersebut. (Beta adalah huruf kedua dari alfabet Yunani.) Awalnya, istilah pengujian alpha berarti tahap pertama pengujian dalam proses pengembangan perangkat lunak. Tahap pertama meliputi pengujian unit, pengujian komponen, dan pengujian sistem. Pengujian beta dapat dianggap "pengujian pra-rilis.

Tujuan pengujian beta adalah menempatkan aplikasi Anda di tangan pengguna nyata di luar tim teknis Anda sendiri untuk menemukan kekurangan atau masalah apa pun dari perspektif pengguna yang tidak ingin Anda miliki dalam versi aplikasi final yang dirilis.



Contoh: Microsoft dan banyak organisasi lain merilis versi beta dari produk mereka untuk diuji oleh pengguna.

Diagram berikut menjelaskan kecocokan pengujian Beta dalam siklus hidup pengembangan perangkat lunak:



## TUGAS PRAKTIKUM 02

1. Pahami dan jelaskan karakteristik perangkat lunak / *software* !
2. Pahami dan jelaskan *Black Box Testing* serta berikan contohnya !
3. Pahami dan jelaskan *White Box Testing* serta berikan contohnya !

Simpan file dengan format **.pdf** dengan nama file **kelas\_nim\_nama\_prak02\_prpl**