



**PROGRAM STUDI
TEKNIK INFORMATIKA**
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

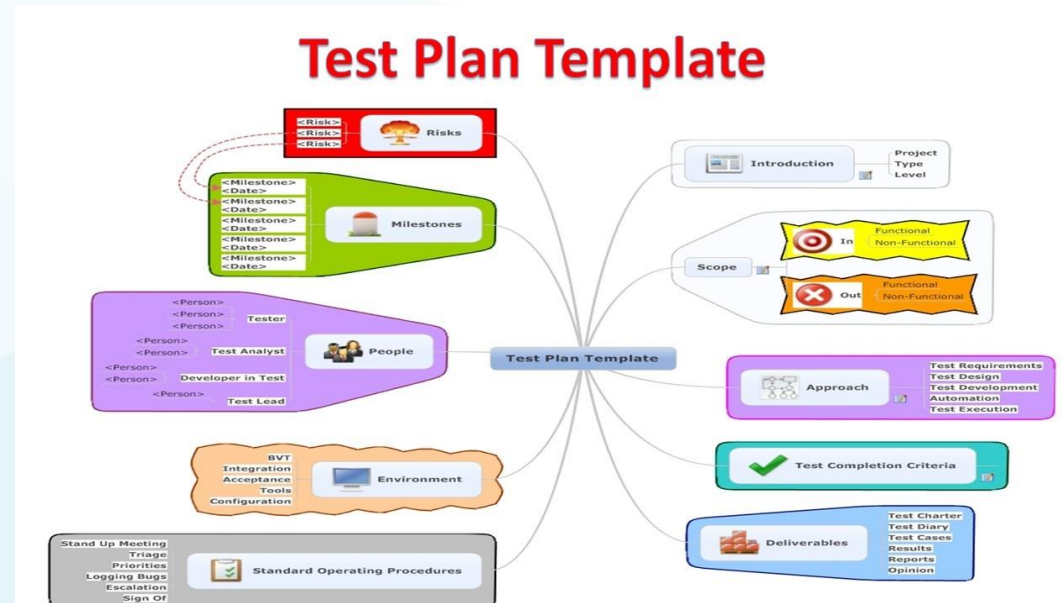
MATA KULIAH
Software Quality and Testing



Software Quality and Testing



Strategi Pengujian Perangkat Lunak



Konten

- 1. Test Plan**
- 2. Pentingnya Test Plan**
- 3. Tahapan Menulis Test Plan**
- 4. Penugasan Test Plan (Kelompok)**

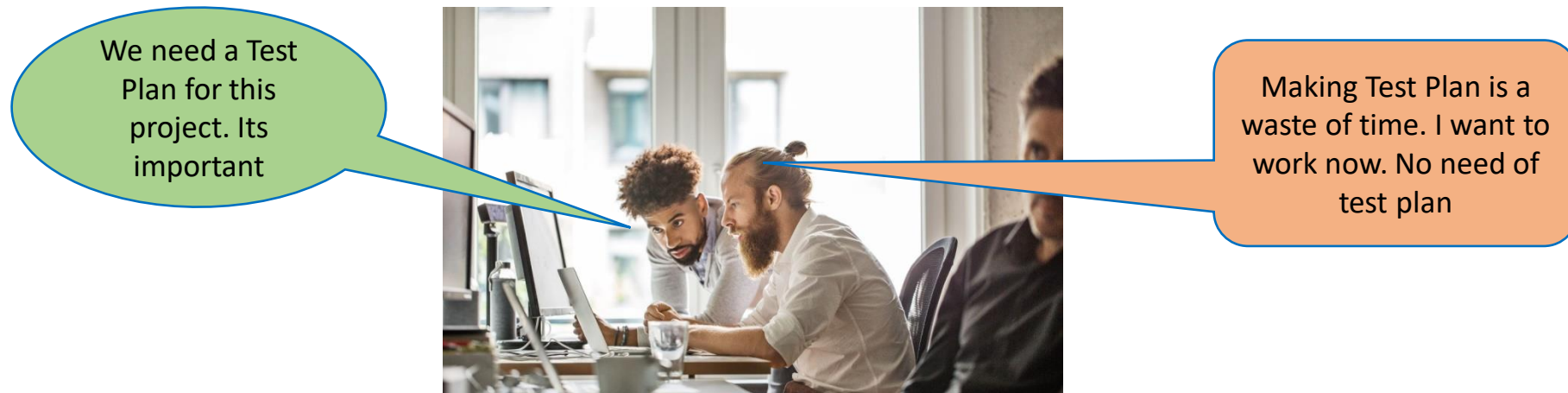
TEST PLAN

Bagaimana menciptakan Rencana Pengujian (Test Plan) ?

Apa itu Test Plan?

Test plan adalah dokumen terperinci yang menguraikan strategi pengujian, tujuan pengujian, sumber daya (tenaga kerja, perangkat lunak, perangkat keras) yang diperlukan untuk pengujian, jadwal pengujian, estimasi pengujian, dan hasil pengujian.

Test plan berfungsi sebagai cetak biru untuk melakukan aktivitas pengujian perangkat lunak sebagai proses yang ditentukan yang dipantau dan dikontrol secara cermat oleh manajer pengujian.



Pentingnya Test Plan

Membuat Rencana Tes memiliki banyak manfaat:

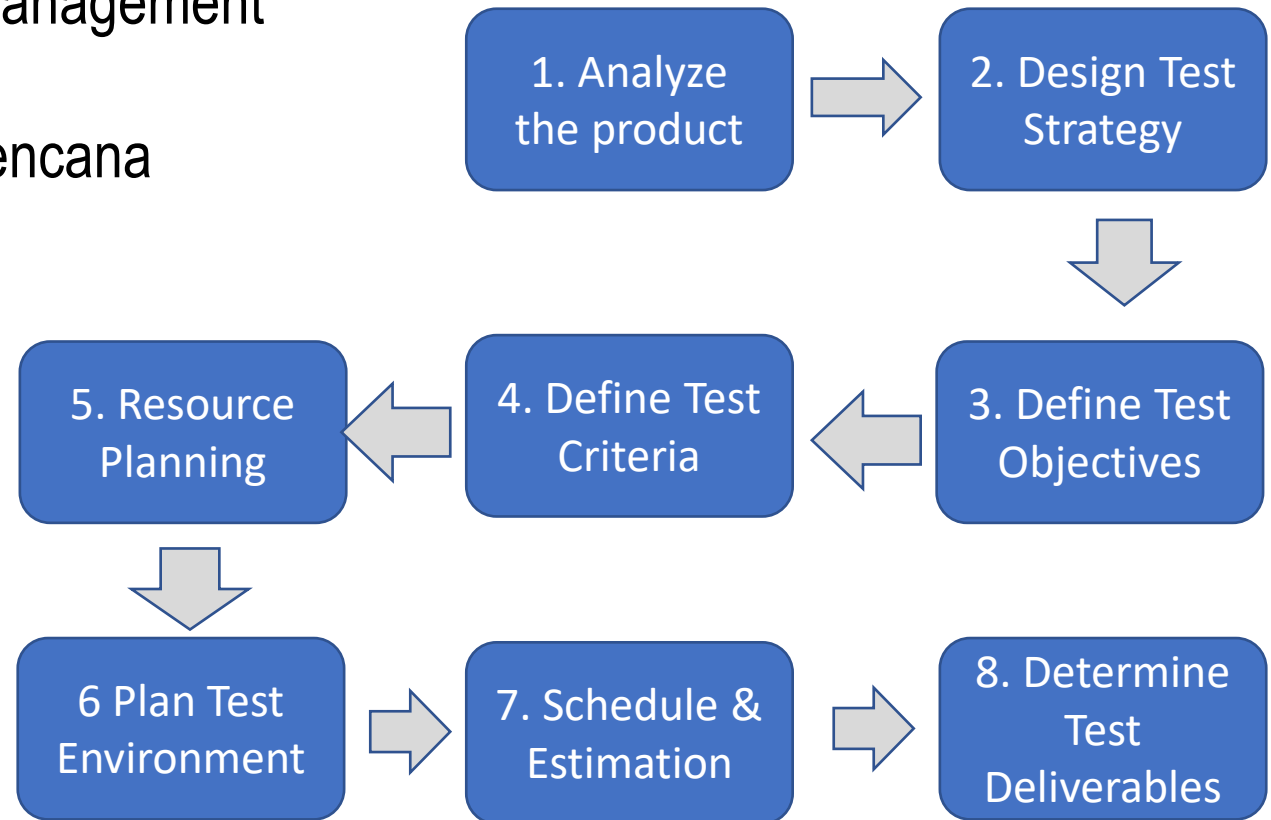
1. Rencana pengujian membantu menentukan **upaya** yang diperlukan untuk mengetahui kualitas aplikasi yang diuji.
2. Membantu orang-orang di luar tim pengujian seperti developer, manajer bisnis, pelanggan **memahami** detail pengujian.
3. Rencana pengujian **memandu** pemikiran kita. Ini seperti buku peraturan, yang perlu diikuti.
4. Aspek penting seperti test estimation, test scope, test strategy **didokumentasikan** dalam Rencana Tes (test plan), sehingga dapat ditinjau oleh Tim Manajemen dan digunakan kembali untuk proyek lain.



Bagaimana cara menulis Test Plan ?

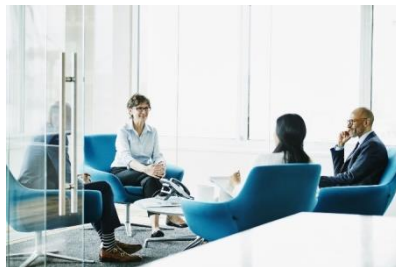
- Test Plan adalah tugas terpenting dari Test Management Process.
- Tujuh langkah di bawah ini untuk membuat rencana pengujian sesuai IEEE 829.

1. Analyze the product
2. Design the Test Strategy
3. Define the Test Objectives
4. Define Test Criteria
5. Resource Planning
6. Plan Test Environment
7. Schedule & Estimation
8. Determine Test Deliverable



Step 1) Analyze the product

- ✓ Bagaimana Anda dapat menguji suatu produk tanpa informasi apa pun tentangnya?
 - Jawabannya adalah Mustahil. Anda harus mempelajari suatu produk secara menyeluruh sebelum mengujinya.
- ✓ Anda harus meneliti klien dan pengguna akhir untuk mengetahui kebutuhan dan harapan mereka dari aplikasi.
 - Siapa yang akan menggunakan situs web ini?
 - Untuk apa itu digunakan?
 - Bagaimana cara kerjanya?
 - Apa perangkat lunak / perangkat keras yang digunakan produk?



Interview
client, designer
and developer



Interview client,
designer and
developer

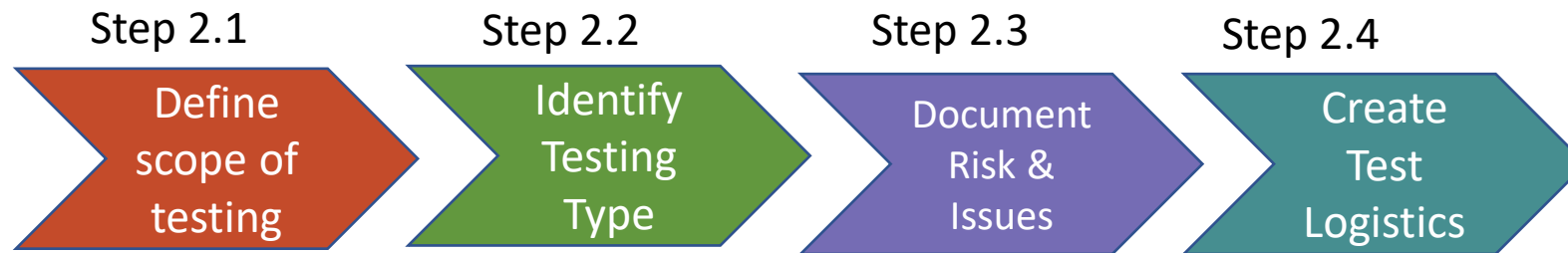


Interview client,
designer and
developer



Step 2) Develop Test Strategy

- **Test Strategy** adalah langkah penting dalam membuat Rencana Tes.
- Dokumen Test Strategy, adalah dokumen tingkat tinggi, yang biasanya dikembangkan oleh Manajer Tes. Dokumen ini menjelaskan:
 - **Tujuan pengujian proyek** dan cara untuk mencapainya.
 - Menentukan **upaya dan biaya** pengujian



Step 2.1) Define Scope of Testing

- Sebelum memulai aktivitas pengujian apa pun, ruang lingkup pengujian harus diketahui. Anda harus memikirkan :
 - Komponen sistem yang akan diuji (perangkat keras, perangkat lunak, middleware, dll.) Didefinisikan sebagai "**dalam cakupan**".
 - Komponen sistem yang tidak akan diuji juga perlu didefinisikan dengan jelas sebagai "**di luar ruang lingkup**".
- Menentukan ruang lingkup proyek pengujian Anda sangat penting bagi semua pemangku kepentingan. Cakupan yang tepat membantu Anda:
 - Berikan semua **orang kepercayaan diri & informasi yang akurat** tentang pengujian yang Anda lakukan.
 - Semua anggota proyek akan memiliki pemahaman **yang jelas** tentang apa yang diuji dan apa yang tidak

Bagaimana Anda menentukan ruang lingkup proyek Anda?

- Untuk menentukan ruang lingkup, kita harus:
 - Persyaratan pelanggan yang tepat
 - Anggaran Proyek
 - Spesifikasi produk
 - Keterampilan & bakat tim uji Anda

Problem Scenario

Pelanggan ingin Anda menguji API-nya. Tetapi anggaran proyek tidak mengizinkan untuk melakukannya. Dalam kasus seperti itu, apa yang akan Anda lakukan?

Dalam kasus seperti itu, Anda perlu meyakinkan pelanggan bahwa Pengujian Api adalah pekerjaan ekstra dan akan menghabiskan sumber daya yang signifikan. Beri dia data yang mendukung fakta Anda. Beri tahu dia jika Pengujian Api termasuk dalam cakupan, anggaran akan meningkat sebesar XYZ.

Pelanggan setuju dan karenanya cakupan baru, item di luar cakupan adalah:

- Item dalam cakupan: Pengujian Fungsional, Pengujian Api
- Item di luar cakupan: Pengujian Basis Data, perangkat keras & antarmuka eksternal lainnya

Step 2.2) Identify Testing Type

- **Testing Type** adalah prosedur pengujian standar yang memberikan hasil pengujian yang diharapkan.
- Setiap jenis pengujian diformulasikan untuk mengidentifikasi jenis bug produk tertentu. Namun, semua Jenis Pengujian ditujukan untuk mencapai satu tujuan yang sama "**Deteksi dini** semua cacat sebelum merilis produk ke pelanggan"
- Jenis **pengujian yang umum** digunakan dijelaskan sebagai berikut:

Unit Test	<ul style="list-style-type: none">• Uji perangkat lunak terkecil yang dapat diverifikasi dalam aplikasi
API Testing	<ul style="list-style-type: none">• Uji API yang dibuat untuk aplikasi tersebut
Integration Test	<ul style="list-style-type: none">• Modul perangkat lunak individu digabungkan dan diuji sebagai satu kelompok
System Test	<ul style="list-style-type: none">• Dilakukan pada sistem yang lengkap dan terintegrasi untuk mengevaluasi kepatuhan sistem dengan persyaratan yang ditentukan
Install/uninstall Testing	<ul style="list-style-type: none">• Fokus pada apa yang pelanggan perlu lakukan untuk menginstal / menghapus instalasi dan mengatur / menghapus perangkat lunak baru dengan sukses
Agile Testing	<ul style="list-style-type: none">• Menguji sistem menggunakan Agile methodology

- Ada **banyak Jenis Pengujian** untuk menguji produk perangkat lunak.
- Tim Anda **tidak dapat memiliki cukup** upaya untuk menangani semua jenis pengujian.
- Sebagai Manajer Tes, Anda harus menetapkan prioritas Jenis Pengujian:
 - Jenis Pengujian mana yang harus **difokuskan** untuk pengujian aplikasi web?
 - Jenis Pengujian mana yang harus **diabaikan** untuk menghemat biaya?

Step 2.3) Document Risk & Issues

- Risiko adalah peristiwa masa depan yang tidak pasti dengan kemungkinan terjadinya dan potensi kerugian. Ketika risiko benar-benar terjadi, itu menjadi 'masalah'.
- Dalam Rencana Pengujian QA, Anda akan mendokumentasikan risiko tersebut

Risk	Mitigation/ Solusi
Anggota tim tidak memiliki keterampilan yang diperlukan untuk pengujian situs web.	Rencanakan kursus pelatihan untuk melatih anggota Anda
Jadwal proyek terlalu ketat; sulit untuk menyelesaikan proyek ini tepat waktu	Tetapkan Prioritas Tes untuk setiap aktivitas tes.
Test Manajer memiliki keterampilan manajemen yang buruk	Rencanakan pelatihan kepemimpinan untuk manajer
Kurangnya kerja sama berdampak negatif pada produktivitas karyawan Anda	Imbullah setiap anggota tim dalam tugasnya, dan ilhami mereka untuk upaya yang lebih besar.
Estimasi anggaran yang salah dan pembengkakan biaya	Tetapkan ruang lingkup sebelum mulai bekerja, berikan banyak perhatian pada perencanaan proyek dan terus melacak dan mengukur kemajuan

Step 2.4) Create Test Logistics

- Dalam Test Logistics, Test Manager harus menjawab pertanyaan berikut:
 - Siapa yang akan menguji?
 - Kapan tes akan dilakukan?
- **Siapa yang akan menguji?**
- Anda mungkin tidak tahu nama pasti penguji yang akan menguji, tetapi jenis penguji dapat ditentukan.

Untuk memilih anggota yang tepat untuk tugas tertentu, Anda harus mempertimbangkan apakah keahliannya memenuhi syarat untuk tugas tersebut atau tidak, juga memperkirakan anggaran proyek. Memilih anggota yang salah untuk tugas tersebut dapat menyebabkan proyek **gagal atau tertunda**.

Orang yang memiliki keterampilan berikut paling ideal untuk melakukan pengujian perangkat lunak:

- Kemampuan untuk memahami sudut pandang pelanggan
- Keinginan kuat akan kualitas
- Perhatian terhadap detail
- Kerjasama yang baik

Dalam proyek Anda, anggota yang akan bertanggung jawab atas pelaksanaan pengujian adalah penguji. Berdasarkan anggaran proyek, Anda dapat memilih anggota sumber atau outsourcing sebagai penguji.

Kapan tes akan dilakukan?

Kegiatan pengujian harus disesuaikan dengan kegiatan pengembangan terkait.

Anda akan mulai menguji saat Anda memiliki semua item wajib yang ditunjukkan pada gambar berikut



*Test specs and
requirement document*



Human resources



Test environment



*Ready to
Test*

Step 3) Define Test Objective

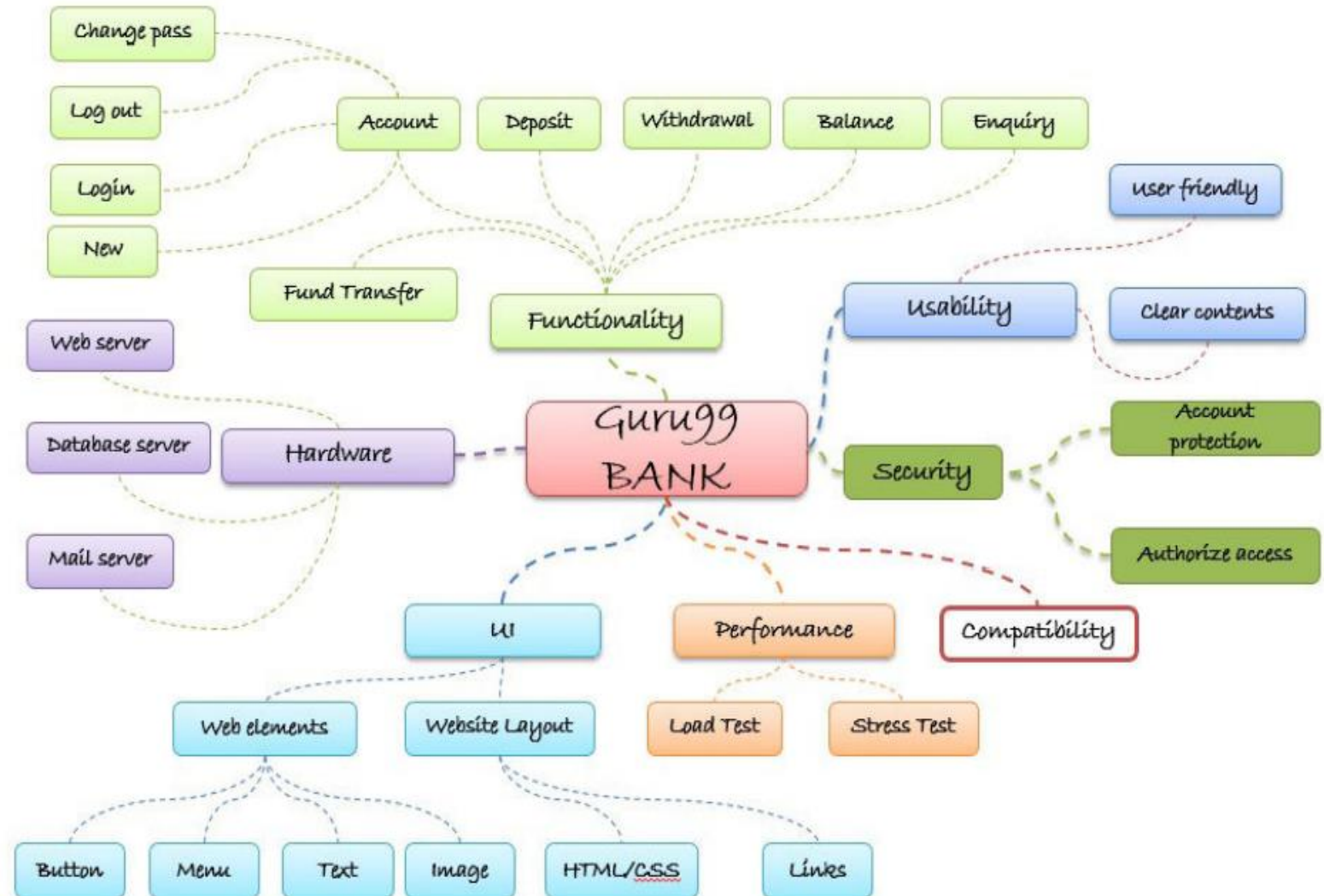
- **Test Objective** adalah tujuan keseluruhan dan pencapaian dari pelaksanaan tes. Tujuan dari pengujian ini adalah menemukan sebanyak mungkin kerusakan perangkat lunak; pastikan perangkat lunak yang diuji bebas bug sebelum dirilis.
- Untuk menentukan tujuan pengujian, Anda harus melakukan 2 langkah berikut
 - Buat daftar semua fitur perangkat lunak (fungsionalitas, kinerja, GUI...) yang mungkin perlu diuji.
 - Tentukan target atau tujuan pengujian berdasarkan fitur di atas

Gambar ini menunjukkan semua fitur yang mungkin dimiliki situs web Guru99. Berdasarkan fitur di atas, Anda dapat menentukan Tujuan Tes proyek Guru99 sebagai berikut:

- Periksa apakah fungsi situs web Guru99 (Akun, Deposit...) berfungsi seperti yang diharapkan tanpa kesalahan atau bug di lingkungan bisnis nyata
- Periksa apakah antarmuka eksternal situs web seperti UI berfungsi seperti yang diharapkan dan & memenuhi kebutuhan pelanggan.
- Verifikasi kegunaan situs web. Apakah fungsionalitas tersebut nyaman bagi pengguna atau tidak?

Website Guru99 Bank:

<http://www.demo.guru99.com/V4/>



Step 4) Define Test Criteria

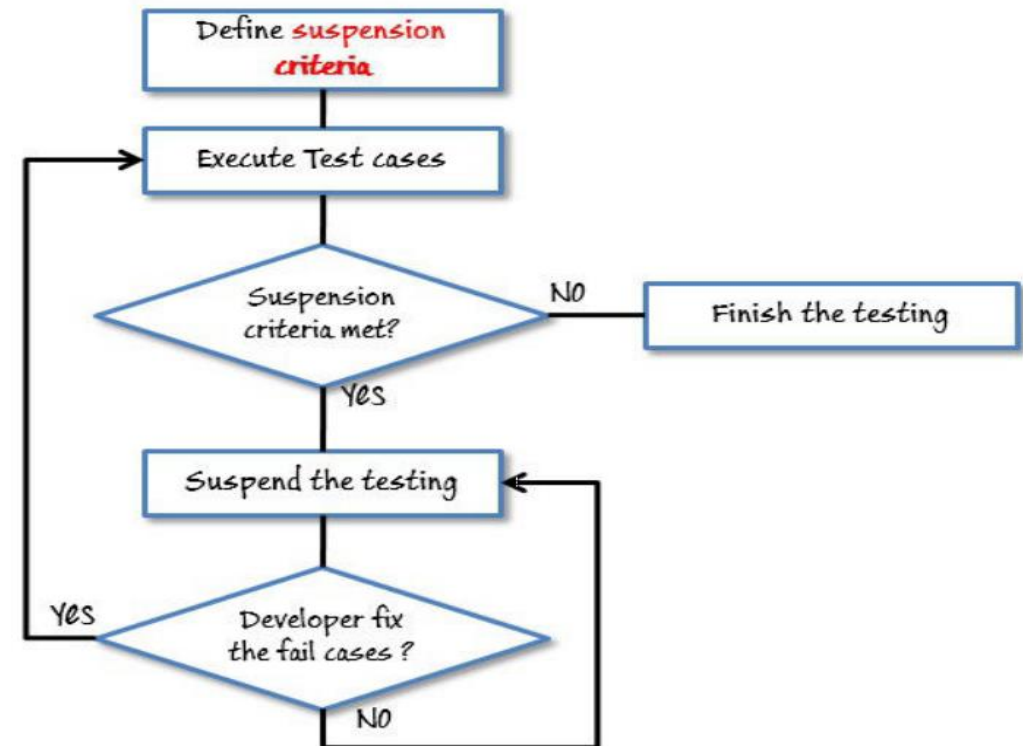
Test Criteria adalah standar atau aturan yang menjadi dasar prosedur tes atau penilaian tes.

Ada 2 jenis kriteria tes sebagai:

1. Kriteria Penangguhan (Suspension Criteria)
2. Kriteria Keluar (Exit Criteria)

Suspension Criteria

- Tentukan kriteria **penangguhan** kritis untuk suatu pengujian. Jika kriteria penangguhan terpenuhi selama pengujian, siklus pengujian aktif akan ditangguhkan hingga kriteria diselesaikan. Contoh: Jika anggota tim Anda melaporkan bahwa ada 40% kasus pengujian yang gagal, Anda harus menangguhkan pengujian hingga tim pengembangan memperbaiki semua kasus yang gagal.



Exit Criteria

- Ini menentukan kriteria yang menunjukkan **berhasil** menyelesaikan fase uji. Kriteria keluar adalah hasil tes yang ditargetkan dan diperlukan sebelum melanjutkan ke tahap pengembangan berikutnya. Contoh: **95%** dari semua kasus pengujian kritis harus lulus.
- Beberapa metode untuk menentukan kriteria keluar adalah dengan menentukan tingkat kelulusan dan tingkat kelulusan yang ditargetkan.
 - Run rate adalah rasio antara jumlah **kasus pengujian yang dijalankan / total kasus pengujian** dari spesifikasi pengujian. Misalnya, spesifikasi pengujian memiliki total 120 TC, tetapi penguji hanya mengeksekusi 100 TC, Jadi run rate adalah $100/120 = 0.83$ (83%)
 - Pass rate adalah rasio antara angka **kasus uji yang lulus / kasus uji yang dijalankan**. Misalnya, di atas 100 TC yang dieksekusi, ada 80 TC yang lulus, jadi tingkat kelulusannya adalah $80/100 = 0,8$ (80%)
- Data ini dapat diambil di dokumen Metrik Uji.
 - Run rate wajib 100% kecuali alasan yang jelas diberikan.
 - Tingkat kelulusan tergantung pada lingkup proyek, tetapi mencapai tingkat kelulusan yang tinggi adalah sebuah tujuan.
- Contoh: Tim Anda telah melakukan eksekusi uji. Mereka melaporkan hasil tes kepada Anda, dan mereka ingin Anda mengkonfirmasi Kriteria Keluar

We have run 90% total of test cases. Remaining 10% test cases we do not how to test

No, I do not confirm

Do you confirm the Exit Criteria ?



Dalam kasus di atas, Run rate wajib diisi 100%, tetapi tim pengujian hanya menyelesaikan 90% dari kasus pengujian. Ini berarti Run rate tidak terpenuhi, jadi JANGAN konfirmasi Exit Criteria

Step 5) Resources Planning

- Rencana sumber daya adalah ringkasan terperinci dari semua jenis sumber daya yang diperlukan untuk menyelesaikan tugas proyek. Sumber daya bisa berupa manusia, peralatan dan bahan yang dibutuhkan untuk menyelesaikan suatu proyek.
- Perencanaan sumber daya merupakan faktor penting dalam perencanaan pengujian karena membantu dalam menentukan jumlah sumber daya (karyawan, peralatan...) yang akan digunakan untuk proyek.
- Oleh karena itu, Test Manager dapat membuat jadwal & estimasi yang tepat untuk proyek tersebut.
- Bagian ini mewakili sumber daya yang direkomendasikan untuk proyek Anda.

Human Resource

No	Member	Tasks
1.	Test Manager	<ul style="list-style-type: none">• Kelola seluruh proyek• Tentukan arah proyek• Dapatkan sumber daya yang sesuai
2.	Tester	<ul style="list-style-type: none">• Mengidentifikasi dan mendeskripsikan teknik pengujian / alat / arsitektur otomatisasi yang sesuai• Verifikasi dan nilai Pendekatan Tes• Jalankan tes, Log hasil, Laporkan cacat.• Penguji dapat berasal dari sumber atau anggota luar, berdasarkan anggaran proyek• Untuk tugas yang membutuhkan keterampilan rendah, saya sarankan Anda memilih anggota outsourcing untuk menghemat biaya proyek.
3.	Developer in Test	Menerapkan test cases, test program, test suite, dll
4.	Test Administrator	<ul style="list-style-type: none">• Membangun dan memastikan Test Lingkungan dan aset dikelola dan dipelihara• SupportTester untuk menggunakan lingkungan pengujian untuk eksekusi pengujian
5.	SQA members	<ul style="list-style-type: none">• Bertanggung jawab atas jaminan kualitas• Periksa untuk memastikan apakah proses pengujian memenuhi persyaratan yang ditentukan

System Resources

- Untuk pengujian, aplikasi web, Anda harus merencanakan sumber daya sebagai tabel berikut:

No	Resources	Descriptions
1.	Server	<ul style="list-style-type: none">• Instal aplikasi web yang sedang diuji• Ini termasuk server web terpisah, server basis data, dan server aplikasi jika berlaku
2.	Test tools	<ul style="list-style-type: none">• Alat pengujian adalah untuk mengotomatiskan pengujian, mensimulasikan operasi pengguna, menghasilkan hasil pengujian.• Ada banyak alat uji yang dapat Anda gunakan untuk proyek ini seperti Selenium, QTP... dll
3.	Network	Anda memerlukan Jaringan yang menyertakan LAN dan Internet untuk mensimulasikan bisnis nyata dan lingkungan pengguna
4.	Computer	PC yang sering digunakan pengguna untuk menghubungkan web server

Step 6) Plan Test Environment

- Apakah Lingkungan Tes itu?

Lingkungan pengujian adalah pengaturan perangkat lunak dan perangkat keras tempat tim pengujian akan menjalankan kasus pengujian. Lingkungan pengujian terdiri dari bisnis nyata dan lingkungan pengguna, serta lingkungan fisik, seperti server, lingkungan berjalan ujung depan.

- Bagaimana cara mengatur Lingkungan Tes?

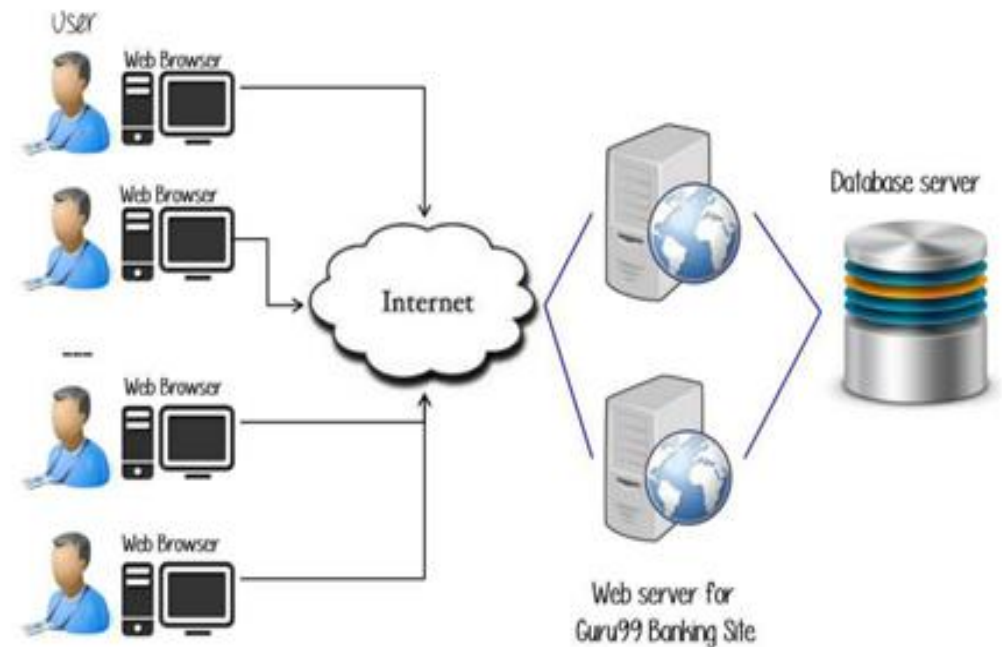
Kembali ke proyek Anda, bagaimana Anda menyiapkan lingkungan pengujian untuk situs web perbankan ini?

Untuk menyelesaikan tugas ini, Anda membutuhkan kerjasama yang kuat antara Tim Penguji dan Tim Pengembang



Anda harus mengajukan beberapa pertanyaan kepada pengembang untuk memahami aplikasi web yang diuji dengan jelas. Berikut beberapa pertanyaan yang direkomendasikan. Tentu saja, Anda dapat mengajukan pertanyaan lain jika perlu.

- Berapa koneksi pengguna maksimum yang dapat ditangani situs web ini pada saat bersamaan?
- Apa persyaratan perangkat keras / perangkat lunak untuk instal situs web ini? Apakah komputer pengguna memerlukan pengaturan khusus untuk menjelajahi situs web?



Step 7) Schedule & Estimation

- Dalam artikel Test Estimation, Anda sudah menggunakan beberapa teknik untuk memperkirakan upaya untuk menyelesaikan proyek. Sekarang Anda harus memasukkan perkiraan tersebut serta jadwal ke Perencanaan Tes
- Dalam tahap Test Estimation, misalkan Anda memecah seluruh proyek menjadi tugas-tugas kecil dan menambahkan estimasi untuk setiap tugas seperti di bawah ini:

Task	Members	Estimate effort
Create the test specification	Test Designer	170 man-hour
Perform Test Execution	Tester, Test Administrator	80 man-hour
Test Report	Tester	10 man-hour
Test Delivery		20 man-hour
Total		280 man-hour

- Kemudian Anda membuat jadwal untuk menyelesaikan tugas ini.

Man-Hour

- 1 man-hour artinya satu orang bekerja selama satu jam.
- Jadi kalau sebuah tugas memiliki 100 man-hour, maka kalau ia dikerjakan satu orang, ia selesai seratus jam.
- Kalau dikerjakan lima orang, selesai dua puluh jam.
- Dikerjakan dua puluh orang, selesai lima jam.

- Membuat jadwal adalah istilah umum dalam manajemen proyek. Dengan membuat jadwal yang solid dalam Test Planning, Test Manajer dapat menggunakannya sebagai alat untuk memantau kemajuan proyek, mengontrol pembengkakan biaya.
- Untuk membuat jadwal proyek, Test Manajer membutuhkan beberapa jenis masukan seperti di bawah ini:
 - **Tenggat waktu karyawan dan proyek:** Hari kerja, tenggat waktu proyek, ketersediaan sumber daya adalah faktor-faktor yang mempengaruhi jadwal
 - **Estimasi proyek:** Berdasarkan estimasi, Manajer Penguji tahu berapa lama waktu yang dibutuhkan untuk menyelesaikan proyek. Sehingga dia bisa membuat jadwal proyek yang sesuai
 - **Risiko Proyek:** Memahami risiko membantu Manajer Penguji menambahkan waktu ekstra yang cukup ke jadwal proyek untuk menangani risiko

Step 8) Test Deliverables

- Hasil pengujian adalah daftar semua dokumen, alat, dan komponen lain yang harus dikembangkan dan dipelihara di support upaya pengujian.
- Ada hasil pengujian yang berbeda di setiap fase siklus pengembangan perangkat lunak.



Hasil pengujian disediakan **sebelum** fase pengujian.

- Dokumen rencana pengujian.
- Dokumen uji kasus
- Spesifikasi Desain Uji.



Hasil pengujian disediakan **selama** pengujian:

- Skrip Tes
- Simulator.
- Uji Data
- Uji Matriks Ketertelusuran
- Log kesalahan dan log eksekusi.



Hasil pengujian diberikan **setelah** siklus pengujian selesai.

- Hasil Tes / laporan
- Laporan Cacat
- Panduan prosedur Instalasi / Pengujian
- Catatan rilis

REFERENSI

- Software Testing : Learn in one day , Krishna Rungta, Guru99, 2019
- Perry, William. Software Testing 2006
- Hendra Jatnika & Yulie Irwan – Testing dan Implementasi Sistem ITB
- Brian Hambling et al., Software Testing, BCS The Chartered Institute for IT, 2010
- Kshirasagar Naik and Priyadarshi Tripathu, Software Testing and Quality Assurance, John Wiley & Sons, Inc., 2008
- Pressman, R.S., 2015, *Software Engineering: A Practitioner's Approach*, Edisi ke-8, Mc Graw-Hill
- Ian Sommerville, *Software Engineering 10th Edition*, Addison-Wesley, 2015

Ada Pertanyaan?



Responsi

Responsi

1. Jelaskan apa yang Anda ketahui tentang Sistem development Life Cycle!
Berikan contoh metode dalam SDLC!
2. Jelaskan apa perbedaan SDLC dengan STLC!
3. Jelaskan apa yang Anda ketahui tentang STLC? Bagaimana kaitannya dengan SDLC?
4. Jelaskan bagaimana strategi pengujian dalam melakukan pengujian Perangkat Lunak yang baik!



Terimakasih

ANY QUESTIONS?