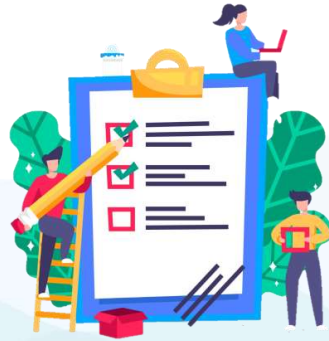




Program Studi Teknik Informatika

FAKULTAS ILMU KOMPUTER  
UNIVERSITAS DIAN NUSWANTORO

MATA KULIAH  
SOFTWARE QUALITY & TESTING



## Teknik Pengujian Perangkat Lunak

<a href="https://www.freepik.com/vectors/people">People vector created by Freepik - www.freepik.com/</a>

FAKULTAS ILMU KOMPUTER

PROGRAM STUDI  
TEKNIK INFORMATIKA

MATA KULIAH  
SOFTWARE QUALITY & TESTING

### Teknik Pengujian Perangkat Lunak

Teknik pengujian perangkat lunak membantu dalam merancang kasus uji yang lebih baik, Ketika pengujian menyeluruh tidak mungkin dilakukan. Teknik pengujian manual membantu mengurangi jumlah kasus uji yang akan dijalankan sambil meningkatkan cakupan pengujian. Hal ini membantu mengidentifikasi kondisi pengujian yang sulit untuk dikenali. Terdapat beberapa Teknik dalam pengujian, diantaranya adalah

- Boundary Value Analysis (BVA)
- Equivalence Class Partitioning
- Decision Table Based Testing
- State Transition
- Error Guessing

FAKULTAS ILMU KOMPUTER

### Boundary Value Analysis (1)

- Analisis Nilai Batas adalah berbasis pengujian pada nilai-nilai batas antara partisi.
- Mencakup maksimum, minimum, di dalam, atau di luar batas, nilai tipikal dan nilai kesalahan.
- Secara umum, bahwa analisis nilai batas didasarkan pada pengujian pada batas-batas antara partisi. Ini mencakup maksimum, minimum, di dalam atau di luar batas, nilai tipikal dan nilai kesalahan. Secara umum terlihat bahwa sejumlah besar kesalahan terjadi di batas dari nilai input yang telah ditentukan.
- Teknik pengujian perangkat lunak ini berdasarkan prinsip bahwa, jika suatu sistem bekerja dengan baik untuk nilai-nilai tertentu, maka itu akan berhasil sangat baik untuk semua nilai yang berada di antara dua batas tersebut

FAKULTAS ILMU KOMPUTER

### Equivalence Class Partitioning (2)

- Partisi Kelas Setara memungkinkan Anda untuk membagi serangkaian kondisi pengujian menjadi partisi yang harus dianggap sama. Perangkat lunak ini metode pengujian membagi domain input dari suatu program ke dalam kelas-kelas data dari mana kasus uji harus dirancang.
- Konsep di balik teknik ini adalah test case dari seorang perwakilan nilai setiap kelas sama dengan tes dari nilai lain yang sama kelas. Ini memungkinkan Anda untuk Mengidentifikasi kesetaraan yang valid dan tidak valid kelas.

## Contoh

- Example: Input conditions are valid between 1 to 10 and 20 to 30
- Hence there are three equivalence classes :
  1. -- to 0 (invalid)
  2. 1 to 10 (valid)
  3. 11 to 19 (invalid)
  4. 20 to 30 (valid)
  5. 31 to --- (invalid)

You select values from each class, i.e., -2, 3, 15, 25, 45

## Decision Table Based Testing (3)

- Tabel keputusan juga dikenal sebagai tabel Sebab-Akibat.
- Perangkat lunak ini teknik pengujian digunakan untuk fungsi yang merespons kombinasi masukan atau kejadian.
- Misalnya, tombol kirim harus diaktifkan jika pengguna telah memasukkan semua bidang yang diperlukan.
- Tugas pertama adalah mengidentifikasi fungsionalitas di mana output bergantung pada kombinasi input.
- Jika ada kumpulan kombinasi input yang besar, kemudian membaginya menjadi himpunan bagian yang lebih kecil yang berguna untuk mengelola a tabel keputusan.
- Untuk setiap fungsi, Anda perlu membuat tabel dan mencantumkan semua jenis types kombinasi input dan outputnya masing-masing.
- Ini membantu untuk mengidentifikasi kondisi yang diabaikan oleh penguji.

## Langkah-langkah Pembuatan Tabel Keputusan

- Mintalah input dalam baris
- Masukkan semua aturan di kolom Isi tabel dengan kombinasi input yang berbeda
- Di baris terakhir, catat output terhadap input kombinasi.

## Contoh Tabel Keputusan

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
Input								
Name	F	T	F	T	F	T	F	T
Email	F	F	T	T	F	F	T	T
Message	F	F	F	F	T	T	T	T
Output								
Submit	F	F	F	F	F	F	F	T

## State Transition (4)

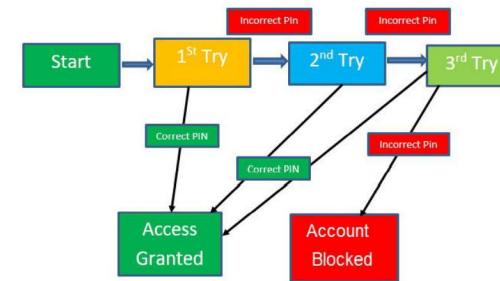
- Dalam teknik state transition, perubahan kondisi input mengubah keadaan Application Under Test (AUT).
- Teknik pengujian ini memungkinkan penguji untuk menguji perilaku AUT.
- Penguji dapat melakukan tindakan ini dengan memasukkan berbagai kondisi input secara berurutan.
- Dalam teknik state transition, tim penguji memberikan masukan positif serta nilai uji masukan negatif untuk mengevaluasi perilaku sistem.

Petunjuk untuk ST :

- Transisi status harus digunakan ketika tim pengujian sedang menguji sistem/aplikasi untuk kumpulan nilai input yang terbatas.
- Teknik ini harus digunakan ketika tim penguji ingin menguji urutan peristiwa yang terjadi dalam aplikasi yang diuji.

## Contoh STD

### State transition diagram



## State Transition Table

	Correct PIN	Incorrect PIN
S1) Start	S5	S2
S2) 1 <sup>st</sup> attempt	S5	S3
S3) 2 <sup>nd</sup> attempt	S5	S4
S4) 3 <sup>rd</sup> attempt	S5	S6
S5) Access Granted	-	-
S6) Account blocked	-	-

## Menebak Kesalahan (Error Guessing) (5)

- Error guessing adalah teknik pengujian perangkat lunak yang didasarkan pada menebak kesalahan yang dapat terjadi dalam kode.
- Ini adalah teknik berbasis pengalaman di mana analis tes menggunakan pengalamannya untuk menebak bagian bermasalah dari aplikasi pengujian.
- Teknik menghitung daftar kemungkinan kesalahan atau situasi rawan kesalahan.
- Kemudian penguji menulis kasus uji untuk mengekspos kesalahan tersebut.
- Untuk merancang tes kasus berdasarkan teknik pengujian perangkat lunak ini, analis dapat menggunakan: pengalaman masa lalu untuk mengidentifikasi kondisi.

**Petunjuk untuk Menebak Kesalahan :**

- Tes harus menggunakan pengalaman pengujian aplikasi sebelumnya yang serupa
- Memahami sistem yang sedang diuji
- Pengetahuan tentang kesalahan implementasi tipikal
- Ingat daerah yang sebelumnya bermasalah
- Evaluasi data Historis & Hasil pengujian Test

**Kesimpulan :**

- Teknik pengujian perangkat lunak memungkinkan kita merancang pengujian yang lebih baik.
- Ada lima teknik utama yang digunakan.
  - Analisis nilai batas adalah pengujian pada batas antara partisi.
  - Partisi Kelas Setara memungkinkan Anda untuk membagi set tes kondisi menjadi partisi yang harus dianggap sama.
  - Teknik pengujian perangkat lunak Tabel Keputusan digunakan untuk fungsi yang merespon kombinasi masukan atau kejadian.
  - Dalam teknik Transisi Keadaan, perubahan kondisi input berubah keadaan Application Under Test (AUT)
  - Tebakan kesalahan adalah teknik pengujian perangkat lunak yang didasarkan pada menebak kesalahan yang dapat terjadi dalam kode.

## ***Matriks Pelacakan Requirement***

**Apakah Matriks Penelusuran ( Traceability Matrix - TM) ?**

- Tracibility matrix adalah dokumen yang menghubungkan dokumen dua baseline apapun yang memerlukan relasi “many to many” untuk memeriksa kelengkapan dari hubungan tersebut.
- Matriks ini digunakan untuk melacak kebutuhan dan untuk memeriksa kebutuhan proyek saat ini apakah sudah terpenuhi.

## Requirement Traceability Matriks (RTM)

- Matriks Penelusuran Kebutuhan mencakup semua kebutuhan yang diusulkan oleh client atau tim pengembang perangkat lunak dan mereka melacak dalam satu dokumen yang dikirim pada akhir siklus hidup pengembangan perangkat lunak.
- Dengan kata lain, RTM adalah dokumen yang memetakan dan melacak kebutuhan user dengan kasus uji.
- Tujuan utama RTM adalah untuk melihat bahwa semua kasus uji tercover sehingga tidak ada fungsi yang harus terlewat saat melakukan pengujian perangkat lunak.

## Mengapa RTM penting ?

- Agenda utama setiap pengujian adalah memahami keinginan klien dan memastikan bahwa produk yang dihasilkan harus bebas cacat.
- Untuk mencapai tujuan ini, setiap QA harus memahami kebutuhan secara menyeluruh dan membuat kasus uji positif dan negatif.
- Ini berarti bahwa kebutuhan perangkat lunak yang disediakan oleh klien harus dibagi lebih lanjut ke dalam skenario yang berbeda dan selanjutnya untuk menguji kasus. Setiap kasus ini harus dieksekusi secara individual.
- Bagaimana memastikan bahwa kebutuhannya diuji dengan mempertimbangkan semua kemungkinan skenario/kasus?
- Bagaimana memastikan kebutuhan apapun tidak ditinggalkan dari siklus pengujian?
- Matriks keterlacakan biasanya merupakan lembar kerja yang berisi: kebutuhan dengan semua kemungkinan skenario dan kasus uji dan keadaan saat ini, yaitu apakah kebutuhan tersebut telah sesuai (passed) atau tidak terpenuhi (gagal). Hal ini membantu tim pengujian untuk memahami tingkat aktivitas pengujian yang dilakukan untuk produk tertentu.

## Parameter pada RTM

- ID kebutuhan
- Jenis dan Deskripsi Kebutuhan
- Kasus Uji dengan Status

Contoh Matriks  
Penelusuran Kebutuhan

Req. No	Rec. Description	ID Kasus Uji	Status
123	Login pada aplikasi	TC01, TC02, TC03	TC01-Pass TC02-Pass TC03-Pass
345	Pembuatan tiket	TC04, TC05, TC06, TC07, TC08, TC09	TC04-Pass TC05-Pass TC06-Fail TC07-No Run
567	Pencarian tiket	TC10, TC11, TC12 TC13, TC14	TC10-Pass TC11-Fail TC12-Pass TC13-No Run

## Project Pengujian Software

No	Req. ID	Req. Desc.	TC. ID	TC. Desc.	Test Design	Test Designer	UAT Test Req ?	Test Execution			Defect	Defect ID	Defect Status	Req. Coverage	Status
								Test Env.	UAT Env.	Prod Env.					
1	Req01	Login ke aplikasi	TC01	login dengan invalid username dan valid password	completed	XYZ	No	Passed	No Run	No Run	None	None	N/A	Partial	
2			TC02	Login dengan valid username dan invalid password	completed	YZA	No	Passed	No Run	No Run	None	None	N/A	Partial	
3			TC03	Login dengan valid credential	completed	XYZ	Yes	Passed	Passed	No Run	Yes	DFCT001	Test Ok	Partial	

## Jenis Matriks Uji Penelusuran

Matriks penelusuran dibagi menjadi 3 komponen utama, yaitu :

### 1. Forward Traceability :

Matriks ini digunakan untuk memeriksa apakah proyek berjalan ke arah yang diinginkan dan produk yang benar. Jenis pengujian FT memastikan bahwa setiap kebutuhan diterapkan pada produk dan bahwa setiap kebutuhan diuji secara menyeluruh. Ini memetakan kebutuhan pada kasus uji.

### 2. Backward or reverse traceability :

Matriks ini digunakan untuk memastikan apakah produk saat ini berada di jalur yang benar. Tujuan di balik jenis ini adalah untuk memverifikasi bahwa kita tidak memperluas cakupan proyek dengan menambah kode, desain elemen, pengujian atau pekerjaan lain yang tidak ditentukan pada kebutuhan. Ini memetakan kasus uji ke kebutuhan.

### 3. Bi-directional traceability (forward + Backward)

Matriks pelacakan ini memastikan bahwa semua kebutuhan tercakup pada kasus uji. Hal ini menganalisis dampak dari perubahan dipengaruhi oleh cacat dalam produk kerja dan sebaliknya.

## Bagaimana Membuat Matriks Penelusuran Kebutuhan ?

- Contoh untuk scenario bahwa customer dapat login pada website perbankan bila input dengan password dan user#id benar.

BR#	Nama Modul	Peran Aplicable	deskripsi
B1	Login dan Logout	Manager Customer	Customer : seorang customer dapat login menggunakan halaman login Manager : seorang manager dapat login menggunakan halaman login customer
B2	Penyelidikan account	Manager Customer	Customer: seorang customer dapat memiliki beberapa rekening bank. Dia dapat melihat jumlah saldo hanya dari rekeningnya Manager : seorang manager dapat melihat rekening dari semua customer yang di bawah supervisinya.
B3	Transfer Dana	Manager Customer	Customer : seorang customer dapat mengirim dana dari rekeningnya ke rekening tujuan. Manager : seorang manager dapat mengirim dana dari beberapa rekening .....

- Step 1 :** Contoh Kasus Uji adalah “ Verifikasi login, ketika user ID dan Password benar dimasukan, ini berhasil”

Kasus Uji #	Kasus Uji	Langkah Uji	Data Uji	Hasil yang diharapkan
1	Verify Login	1) Go to login page 2) Enter userID 3) Enter Password 4) Click Login	Id=Budi89 Pass = 1234	Login successful

Bila ID dan password benar dimasukkan maka harusnya login sukses

- Step 2 :** Mengidentifikasi kebutuhan teknis bahwa kasus uji ini diverifikasi. Untuk kasus uji, kebutuhan teknis adalah T94 sedang diverifikasi

**T94** Jika userID dan password adalah valid. Login

T94 adalah kebutuhan teknis yang memverifikasi login sukses

- Step 3 :** Catat Kebutuhan Teknikal (T94) pada kasus uji.

Kasus Uji #	TR#	Kasus Uji	Test Steps	Test Data	Expected
1	T94	Verify Login	1) Go to login Page 2) Enter UserID 3) Enter Password 4) Click Login	Id=Budi89 Pass=1234	Login successful

Catat Kebutuhan Teknikal pada Kasus Uji

- Step 4.** Mengidentifikasi Kebutuhan Bisnis untuk TR (Technical Requirement-T94) yang didefinisikan.

BR #	Nama Modul	Peran Aplikasi	Deskripsi
B1	Login dan Logout	Manager Customer	Customer : seorang customer dapat login menggunakan halaman login Manager : seorang manager dapat login menggunakan halaman login customer. Homepage login yang di post akan menampilkan link yang berbeda berdasarkan perannya.

Mengidentifikasi Business Requirement dimana T94 didefinisikan

- Step 5. Catat BR (Business Requirement) pada kasus uji

Kasus Uji #	BR #	TR#	Kasus Uji	Test Steps	Test Data	Expected
1	B1	T94	Verify Login	1) Go to login Page 2) Enter UserId 3) Enter Password 4) Click Login	Id=Budi89 Pass=1234	Login successful

- Step 6. lakukan semua kasus uji. Terakhir inti dari 3 kolom pertama dari rangkaian pengujian.

Business Requirement #	Technical Requirement #	Test Case ID
B1	T94	1
B2	T95	2
B3	T96	3
B4	T97	4

Requirement  
Traceability  
matrix

## Keuntungan Matriks Penelusuran Kebutuhan (RTM)

- RTM mengkonfirmasi cakupan pengujian 100%
- RTM menyoroti kebutuhan yang hilang atau dokumen yang tidak konsisten
- RTM menampilkan cacat keseluruhan atau status eksekusi dengan focus pada kebutuhan bisnis.
- RTM membantu dalam menganalisis atau memperkirakan dampak pada tim QA bekerja dengan meninjau Kembali atau mengerjakan ulang kasus uji.

## Lingkungan Pengujian

## Menyiapkan Lingkungan untuk Pengujian Perangkat Lunak

- Menyiapkan Lingkungan pengujian adalah pengaturan software dan hardware untuk tim pengujian dalam melaksanakan kasus uji. Hal ini mendukung eksekusi pengujian dengan hardware dan software, dan jaringan yang dikonfigurasi.
- Tempat pengujian dan lingkungan pengujian dikonfigurasi sesuai kebutuhan aplikasi yang diuji (AUT).
- Pada beberapa kesempatan, tempat pengujian dapat dipadukan/ dikombinasikan menjadi lingkungan pengujian dan data pengujian yang dioperasikan.
- Menyiapkan lingkungan pengujian yang tepat memastikan keberhasilan pengujian perangkat lunak. Setiap kekurangan dalam proses ini dapat menyebabkan biaya dan tambahan waktu bagi client.

## Area Utama pada Pengujian Environment

- Sistem dan aplikasi
- Data uji
- Server database
- Lingkungan yang berjalan front-end
- Sistem operasi klien
- Browser
- Perangkat keras termasuk server sistem Operasi
- Jaringan
- Dokumentasi yang diperlukan seperti dokumen referensi/ petunjuk konfigurasi/ petunjuk instalasi/ user manual

## Proses Pengaturan Lingkungan Pengujian Perangkat Lunak

- Tes terbatas pada apa yang dapat diuji dan apa yang tidak harus diuji.
- Orang-orang yang terlibat dalam setup lingkungan pengujian:
  - Admin Sistem,
  - Developer
  - Tester
  - Terkadang pengguna atau teknisi yang tergabung dalam pengujian.

## Pengaturan berbagai area yang berbeda Uji Lingkungan

- Lingkungan pengujian memerlukan pengaturan dari berbagai nomor dari area yang berbeda seperti,
  - Pengaturan Server Uji : Untuk setiap tes yang tidak dapat dijalankan pada mesin local, mungkin perlu membangun server uji, yang dapat mendukung aplikasi.
- Jaringan: setup internet, wifi LAN, jaringan pribadi, hal ini untuk memastikan bahwa kemacetan yang terjadi selama pengujian tidak berdampak pada anggota lainnya ( developer, desainer, penulis konten, dll)
- Pengaturan pengujian PC: untuk pengujian web mungkin perlu menyiapkan browser yang berbeda untuk pengujian yang berbeda. Sedangkan untuk desktop, memerlukan jenis OS untuk PC yang berbeda.
- Pelaporan Bug ( tool pelaporan Bug harus disediakan untuk pengujian)

## Pembuatan Data Uji untuk Lingkungan Uji

- Pendekatan umum yang digunakan adalah menyalin data produksi ke data uji. Ini membantu pengujian, untuk mendeteksi masalah yang sama seperti server produksi secara langsung tanpa merusak data produksi.
- Pendekatan untuk menyalin data produksi ke data uji, meliputi :
  - Siapkan pekerjaan produksi untuk menyalin data ke lingkungan pengujian yang umum.
  - Semua PII (Personal Identifiable Information) dimodifikasi Bersama dengan data sensitive lainnya. PII diganti dengan data yang benar secara logis, tetapi bukan data pribadi
  - Pindahkan/ hapus data yang tidak relevan dengan pengujian kita.
- Penguji atau pengembang dapat menyalin ini ke lingkungan uji individu mereka. Mereka dapat memodifikasi sesuai dengan kebutuhannya.
- Privasi adalah masalah utama dalam pengkopian data produksi. Untuk mengatasinya, kita harus melihat data uji yang dikaburkan dan dianonimkan.
- Pendekatan Anonimisasi data :
  - Blacklist : semua field data dibiarkan tidak berubah, kecuali field/ item yang ditentukan oleh pengguna
  - Whitelist : secara default, pendekatan ini menganonimkan semua field data, kecuali untuk daftar field yang diijinkan untuk disalin



## Pengelolaan Lingkungan Pengujian

Manajemen Lingkungan Uji berkaitan dengan pemeliharaan dan merapikan tempat uji.

Daftar Aktivitas oleh fungsi manajemen lingkungan Uji, meliputi :

1. Pemeliharaan repositori pusat dengan memperbaharui semua versi lingkungan pengujian
2. Manajemen lingkungan pengujian sesuai permintaan tim pengujian.
3. Sesuai dengan kebutuhan baru yang menciptakan lingkungan baru
4. Pemantauan lingkungan
5. Memperbarui/menghapus lingkungan pengujian yang kedaluwarsa
6. Investigasi masalah lingkungan
7. Koordinasi sampai penyelesaian masalah.

## Checklist Lingkungan Pengujian

Hardware		
1	Apakah peralatan yang dibutuhkan untuk pengujian tersedia ?	Jika ya tidak terjadi, analisis pasokan waktu !
	Cek apakah peralatan peripheral tersedia ?	seperti scanner, printer khusus, dll
Software/ Koneksi		
2	Apakah diperlukan aplikasi khusus ?	seperti excel, word, drawing, dll
	Untuk perangkat lunak baru apakah ada lingkungan pengujian untuk organisasi ?	Apakah organisasi memiliki pengalaman menggunakan dan memelihara perangkat lunak ?
Data Lingkungan		
3	Cek apakah data uji standar tersedia ?	dengan set uji regresi, pertimbangkan administrasi cetak untuk mengoptimalkan data uji
	Apakah ada kesepakatan dengan pemilik data uji tentang data uji yang tersedia ?	pertimbangkan maintenance fungsional
Tool Maintenance/ proses		
4	Cek apakah tersedia satu kontak untuk pemeliharaan lingkungan uji ?	Jika tidak, siapkan daftar dari semua kemungkinan member yang terlibat dalam penjagaan berjalannya lingkungan pengujian
	Apakah kesepakatan yang dicapai tentang keadaan dan kualitas lingkungan pengujian ?	mialnya, kriteria penerimaan, kebutuhan maintenance, dll. juga cek apakah atribut lain atau kualitas ekstra untuk lingkungan sudah sesuai ?

## Bagaimana menyiapkan Manajemen Lingkungan Pengujian yang baik ?

1. Pahami kebutuhan pengujian secara menyeluruh dan pelajari anggota tim pengujian.
2. Konektivitas harus diperiksa sebelum memulai pengujian
3. Periksa perangkat keras dan perangkat lunak yang diperlukan, lisensi
4. Browser dan versi
5. Perencanaan Jadwal menggunakan lingkungan pengujian.
6. Peralatan otomatisasi dan konfigurasinya.

## Kesimpulan

- Lingkungan pengujian adalah pengaturan perangkat lunak dan perangkat keras dimana tim penguji akan melakukan pengujian
- Untuk lingkungan pengujian, area utama yang harus disiapkan meliputi:
  - Sistem dan aplikasi
  - Data percobaan
  - Server basis data
  - Lingkungan berjalan front-end, dll.
- Beberapa tantangan saat menyiapkan lingkungan pengujian meliputi,
  - Lingkungan tersembunyi/ terpendek
  - Penggunaan gabungan antar tim
  - Waktu penyiapan yang rumit
  - Perencanaan yang tidak efektif untuk penggunaan sumber daya untuk integrasi
  - Konfigurasi pengujian yang kompleks

## Test Data Generation ( Apa, bagaimana, contoh, dan tools)

- Pengujian adalah proses yang menghasilkan dan membutuhkan data dalam jumlah yang sangat besar. Data yang digunakan dalam pengujian menggambarkan kondisi awal untuk pengujian dan mewakili media yang digunakan untuk pengujian dalam mempengaruhi perangkat lunak.
- Apa itu Data Uji ?
- Mengapa menggunakan Data Uji ?

## Data Uji ?

- Data uji adalah input yang diberikan ke program perangkat lunak. Data uji mewakili data yang mempengaruhi atau dipengaruhi oleh eksekusi modul tertentu.
- Beberapa data dapat digunakan untuk pengujian positif, biasanya untuk memverifikasi apakah sekumpulan input yang diberikan ke fungsi yang diberikan menghasilkan Hasil yang diharapkan.
- Data lain dapat digunakan untuk pengujian negatif untuk menguji kemampuan program untuk menangani yang tidak biasa, ekstrim, luar biasa, atau masukan yang tidak terduga.
- Data pengujian yang dirancang dengan buruk mungkin tidak menguji semua kemungkinan skenario pengujian yang akan menghambat kualitas perangkat lunak.



- Apakah Pembangkitan Data Uji ?
- Mengapa data uji harus dibuat sebelum mengeksekusi pengujian ?

## DATA UJI

Mengapa Data Uji harus dibuat sebelum pengujian dilakukan ?

Hal ini tergantung dari lingkungan pengujiannya, bila perlu menciptakan data uji (sebagian besar waktu) atau paling tidak mengidentifikasi data uji yang sesuai dengan kasus uji yang sudah dibuat.

- Data uji (biasanya) dibuat sinkron dengan kasus uji yang akan digunakan. Data uji dapat dihasilkan melalui beberapa cara, yaitu :
  1. Secara Manual
  2. Copy Data dari hasil lingkungan pengujian
  3. Copy Data Uji dari system klien
  4. Alat pembuatan data Uji otomatis
- Biasanya data sampel harus dibuat sebelum memulai mengeksekusi pengujian karena sulit untuk menangani manajemen data pengujian.

## Beberapa Kebutuhan Data Uji untuk Jenis Pengujian

- Data Uji untuk Pengujian White Box
- Data Uji untuk Pengujian Kinerja Sistem
- Data Uji untuk Pengujian Keamanan
- Data Uji untuk Pengujian Black Box

## Data Uji untuk Pengujian White Box

Dalam Pengujian White Box, Manajemen data uji diturunkan dari pemeriksaan code yang akan diuji secara langsung.

Data Uji dapat dipilih dengan memperhatikan hal-hal berikut di bawah ini :

- Diharapkan untuk menutupi sebanyak mungkin cabang; data pengujian dapat dihasilkan sedemikian rupa sehingga semua cabang di sumber program kode diuji setidaknya sekali
- Pengujian jalur: semua jalur dalam kode sumber program diuji setidaknya sekali - persiapan data uji dapat dilakukan untuk mencakup banyak kasus

## Data Uji untuk Pengujian Kinerja Sistem

- Pengujian Kinerja adalah jenis pengujian yang dilakukan untuk menentukan seberapa cepat sistem merespons di bawah beban kerja tertentu.
- Tujuan dari jenis pengujian ini bukan untuk menemukan bug, tetapi untuk menghilangkan kemacetan.
- Sebuah aspek penting dari Pengujian Kinerja adalah bahwa kumpulan data sampel yang digunakan harus sangat dekat dengan data 'nyata' atau 'langsung' yang digunakan pada produksi.
- Pelanggan dapat memberikan beberapa data yang sudah mereka miliki atau, jika mereka tidak memiliki kumpulan data yang ada, mereka dapat membantu dengan memberikan umpan balik tentang bagaimana data dunia nyata mungkin terlihat.
- Jika kita berada dalam proyek pengujian pemeliharaan, kita dapat menyalin data dari lingkungan produksi ke lingkungan pengujian. Hal ini bagus untuk menganonimkan (mengacak) data pelanggan yang sensitif seperti Sosial Nomor Keamanan, Nomor Kartu Kredit, Rincian Bank, dll

## Data Uji untuk Pengujian Keamanan

- Pengujian Keamanan adalah proses yang menentukan apakah suatu informasi sistem melindungi data dari niat jahat.
- Kumpulan data yang dibutuhkan dirancang untuk sepenuhnya menguji keamanan perangkat lunak harus mencakup hal berikut ini :
  - **Kerahasiaan:** Semua informasi yang diberikan oleh klien disimpan dalam kerahasiaan yang ketat dan tidak dibagikan dengan pihak luar mana pun semua Pihak. Sebagai contoh, jika aplikasi menggunakan SSL, dapat merancang satu set data uji yang memverifikasi bahwa enkripsi dilakukan benar.
  - **Integritas:** Menentukan bahwa informasi yang diberikan oleh sistem sudah benar. Untuk merancang data uji yang sesuai, kita dapat memulai dengan melihat secara mendalam pada desain, kode, database, dan file struktur.
  - **Otentikasi:** Merupakan proses membangun identitas seorang pengguna. Data pengujian dapat dirancang sebagai yang berbeda kombinasi nama pengguna dan kata sandi dan tujuannya adalah untuk memeriksa apakah hanya orang yang berwenang yang dapat mengakses sistem perangkat lunak.
  - **Otorisasi:** Memberi tahu apa hak pengguna tertentu. Data pengujian mungkin berisi kombinasi yang berbeda dari pengguna, peran dan operasi untuk memeriksa hanya pengguna dengan cukup hak istimewa dapat melakukan operasi tertentu.

## Data Uji untuk Pengujian Black Box

- Dalam Pengujian Black Box, kode tidak terlihat oleh penguji. Kasus uji fungsional dapat memiliki data uji yang memenuhi kriteria berikut :
  - **Tidak ada data:** Periksa respons sistem saat tidak ada data yang dikirimkan
  - **Data valid:** Periksa respons sistem ketika data uji yang valid adalah diserahkan
  - **Data tidak valid:** Periksa respons sistem saat data pengujian Tidak Valid diserahkan
  - **Format data ilegal:** Periksa respons sistem saat data uji masuk format yang tidak valid
  - **Dataset Kondisi Batas:** Data uji memenuhi kondisi nilai batas
  - **Data set Partisi Ekuivalensi:** Data uji yang memenuhi syarat partisi kesetaraan.
  - **Data set Tabel Keputusan:** Data uji yang memenuhi syarat strategi pengujian table keputusan
  - **Data set Uji Transisi Status:** Data uji memenuhi strategi pengujian state transisi
  - **Data Uji Use Case :** Data Uji sinkron dengan use case.

## Pengujian Statik

## Pengujian Statik ?

- Pengujian statik didefinisikan sebagai pengujian perangkat lunak yang digunakan untuk memeriksa cacat pada perangkat lunak tanpa benar-benar menjalankannya.
- Bagian penghitungannya adalah pengujian dinamis yang memeriksa aplikasi Ketika program dijalankan.
- Pengujian statis dilakukan untuk menghindari kesalahan pada tahap awal pengembangan karena lebih mudah Ketika menemukan sumber kegagalannya, dan kegagalannya itu sendiri.
- Pengujian statis membantu menemukan kesalahan yang mungkin tidak ditemukan oleh pengujian dinamis. Dua tipe utama dari pengujian statis adalah :
  1. Pengujian Manual
  2. Analisis otomatis menggunakan tool

## Apakah Review Pengujian Statik ?

Review dalam pengujian statis adalah proses atau pertemuan yang diadakan untuk menemukan cacat yang berpotensi dalam perancangan program apapun. Selain itu juga review digunakan agar semua anggota tim mengetahui tentang kemajuan proyek, dengan memahami bersama harapannya pemikiran yang beragam dapat menghasilkan saran yang sangat bagus. Dokumen langsung diperiksa dan perbedaan dipilah-pilah. Ulasan selanjutnya dapat diklasifikasikan ke dalam lima bagian, yaitu :

1. Review informal
2. Panduan
3. Review teknis
4. Inspeksi
5. Analisis Statik
  1. Data Flow
  2. Kontrol flow

### Tipe Partisipan pada Proses Review

1. **Moderator:** Melakukan pemeriksaan masukan, menindaklanjuti pengerjaan ulang, melatih anggota tim, menjadwalkan pertemuan.
2. **Penulis:** Bertanggung jawab untuk memperbaiki cacat yang ditemukan dan meningkatkan kualitas dokumen
3. **Scribe:** Ia melakukan pencatatan cacat selama peninjauan dan menghadiri rapat peninjauan
4. **Peninjau:** Periksa bahan untuk cacat dan periksa (defect dan inspect)
5. **Manajer:** Memutuskan pelaksanaan tinjauan dan memastikan tujuan proses peninjauan terpenuhi.

### Jenis Cacat yang ditemukan selama pengujian Statis :

1. Penyimpangan dari standar
2. Kode yang tidak dapat dipelihara
3. Cacat desain
4. Persyaratan tidak ada
5. Spesifikasi antarmuka yang tidak konsisten

Biasanya, cacat yang ditemukan selama pengujian statis disebabkan oleh keamanan kerentanan, variabel yang tidak dideklarasikan, pelanggaran batas, sintaks pelanggaran, antarmuka yang tidak konsisten, dll.

### Mengapa Pengujian Statis ?

Pengujian statis dilakukan karena alasan berikut:

- Deteksi dan koreksi cacat dini
- Mengurangi rentang waktu pengembangan
- Mengurangi biaya dan waktu pengujian
- Untuk peningkatan produktivitas pembangunan
- Untuk mendapatkan lebih sedikit cacat pada tahap pengujian selanjutnya

### Apakah yang diuji pada Pengujian Statis ?

- Unit Test Cases
- Business Requirements Document (BRD)
- Use Cases
- System/Functional Requirements
- Prototype
- Prototype Specification Document
- DB Fields Dictionary Spreadsheet
- Test Data
- Traceability Matrix Document
- User Manual/Training Guides/Documentation
- Test Plan Strategy Document/Test Cases
- Automation/Performance Test Scripts

## Bagaimana Pengujian Statis dilakukan ?

Untuk melakukan Static Testing dilakukan dengan cara sebagai berikut:

- Lakukan proses inspeksi untuk sepenuhnya memeriksa desain inspect dari aplikasi
- Gunakan daftar yang diperiksa untuk setiap dokumen yang sedang ditinjau untuk memastikan semua ulasan tercakup sepenuhnya.

## Aktivitas untuk melakukan Pengujian Statis

1. **Validasi Persyaratan Kasus Penggunaan:** Validasi ini bahwa semua tindakan pengguna akhir diidentifikasi, serta input dan output apa pun terkait dengan mereka. Semakin detail dan menyeluruh penggunaannya kasus, semakin akurat dan komprehensif kasus uji dapat menjadi.
2. **Validasi Persyaratan Fungsional:** Ini memastikan bahwa Persyaratan Fungsional mengidentifikasi semua elemen yang diperlukan. Juga melihat fungsionalitas database, daftar antarmuka, dan kebutuhan perangkat keras, perangkat lunak, dan jaringan.
3. **Tinjauan Arsitektur:** Semua proses tingkat bisnis seperti server lokasi, diagram jaringan, definisi protokol, penyeimbangan beban, aksesibilitas basis data, peralatan uji, dll.
4. **Validasi Prototipe/Layar Mockup:** Tahap ini meliputi validasi persyaratan dan kasus penggunaan.
5. **Validasi Kamus Bidang:** Setiap bidang di UI didefinisikan cukup baik untuk membuat kasus uji validasi tingkat lapangan. Bidang adalah periksa panjang min/maks, nilai daftar, pesan kesalahan, dll.

## Tips untuk Keberhasilan Proses Pengujian Statis

Beberapa tip berguna untuk melakukan proses pengujian statis di Rekayasa Perangkat Lunak :

- Fokus hanya pada hal-hal yang benar-benar diperhitungkan
- Rencanakan dan lacak aktivitas peninjauan secara eksplisit. Panduan perangkat lunak dan inspeksi umumnya digabungkan menjadi ulasan sejawat
- Latih peserta dengan Contoh
- Menyelesaikan masalah orang
- Jaga agar proses tetap formal sebagai budaya proyek
- Perbaikan Berkelanjutan – Proses dan Alat
- Dengan menghilangkan penundaan utama dalam pelaksanaan pengujian, biaya pengujian dan waktu bisa dikurangi

## Kesimpulan

- Pengujian statis adalah untuk menemukan cacat sedini mungkin.
- Pengujian statis bukan pengganti pengujian dinamis, keduanya menemukan berbagai jenis cacat
- Review adalah teknik yang efektif untuk Pengujian Statis
- Review tidak hanya membantu menemukan cacat tetapi juga memahami kekurangan kebutuhan, cacat desain, kode yang tidak dapat dipelihara,