

Workshop Kualitas Perangkat Lunak

Teknik Testing

Testing




01

Whitebox Testing

02


Blackbox Testing





Teknik Testing Whitebox Texting


Basis Path Testing



Flow Graph

- Notasi sederhana yang merepresentasikan aliran control
- Proses dan keputusan yg berurutan dimapping menjadi 1 node

Cyclomatic Complexity

- Angka yang menyatakan jumlah jalur independen/jalur dasar dari sebuah program
 - Menunjukkan jumlah pengujian (kasus uji) yang harus dieksekusi
 - setiap jalur dalam program yang memiliki setidaknya satu set pernyataan (processing statement) atau satu kondisi yang baru sama sekali (blm digunakan oleh jalur sebelumnya)
- 

Teknik Testing Whitebox Texting

Graph Martic



Graph Matrix

Item atau event darstruktur data yang dapat membantu dalam mengembangkan alat untuk otomatisasi pengujian jalur.i software system yang dapat diverifikasi dengan test case



Teknik Testing Blackbox Texting



Boundary Value Analysis (BVA)

Equivalence Class Partitioning

Decision Table Based Testing



Teknik Testing Blackbox Texting

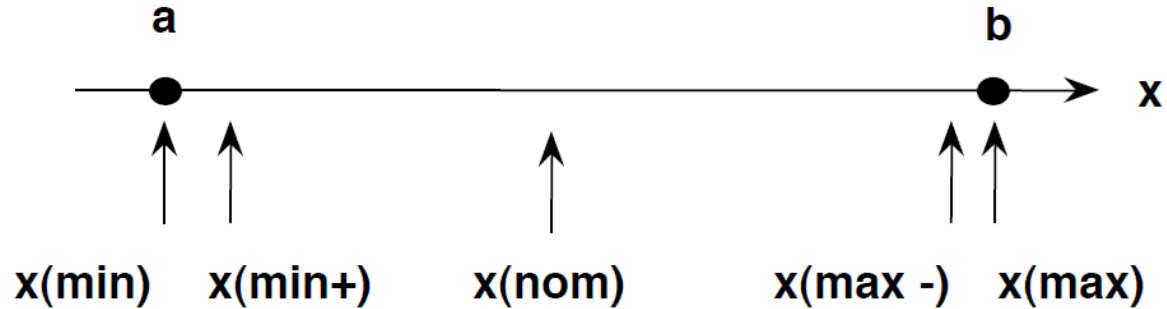


State Transition

Error Guessing

Use case testing

Boundary Value Analysis (BVA)



Contoh:

- Input value valid ketika nilai antara 1 hingga 10
- Boundary valuenya adalah: 0,1,2, 5 dan 9,10,11



Equivalence Class Partitioning



- Kondisi input valid antara 1-10 dan 20-30
- Dan 5 class equivalence nya adalah:
 - --- hingga 0 (invalid)
 - 1 hingga 10 (valid)
 - 11 hingga 19 (invalid)
 - 20 hingga 30 (valid)
 - 31 hingga --- (invalid)
- Kemudian dipilih dari tiap kelasnya:
 - -2, 3, 15, 25, 45



Decision Table Based Testing

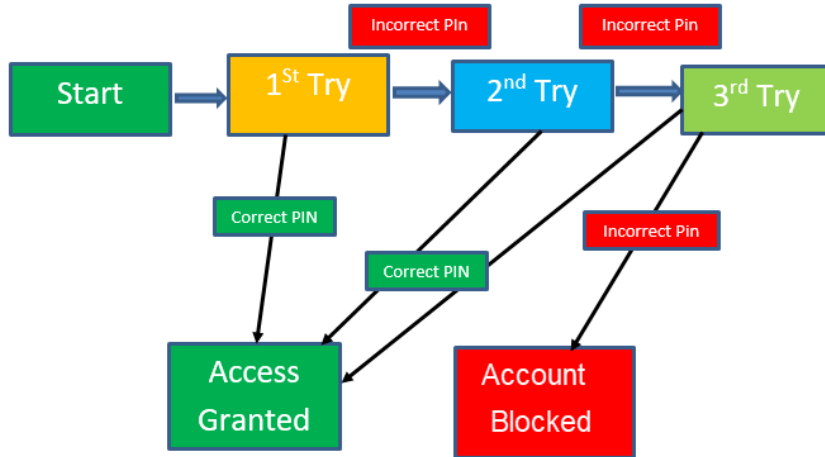


Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Username (T/F)	F	T	F	T
Password (T/F)	F	F	T	T
Output (E/H)	E	E	E	H

- T – Correct username/password
- F – Wrong username/password
- E – Error message is displayed
- H – Home screen is displayed



State Transition Testing



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



Error Guessing



Guidelines untuk Error Guessing:

- Test harus menggunakan pengalaman sebelumnya dengan software yang serupa.
- Mengerti sistem yang akan dites
- Memahami implementasi eror yang ada
- Mengenal area yang sering terjadi eror
- Mengevaluasi data historical dan hasil tes



Use case testing



Main Success Scenario	Step	Description
A:Actor S:System	1	A: Enter Agent Name & Password
	2	S: Validate Password
	3	S: Allow Account Access
Extensions	2a	<u>Password not valid</u> S: Display Message and ask for re-try 4 times
	2b	<u>Password not valid 4 times</u> S: Close Application



Workshop Kualitas Perangkat Lunak

Manual vs Automation

Manual Testing



- Test secara manual tanpa bantuan otomatisasi (tools ataupun scripts) dan dikerjakan langsung oleh manusia.
- Manual testing digunakan tanpa perlu memahami testing tools.
- Software Testing Fundamental salah satunya adalah "**100% Automation is not possible**".



Tujuan Manual Testing



- Test case dibuat saat fase testing dan menyeluruh hingga 100%.
- Juga memastikan bahwa cacat yang dilaporkan diperbaiki oleh pengembang dan pengujian ulang telah dilakukan oleh penguji pada cacat tetap.
- Pada dasarnya, pengujian ini memeriksa kualitas sistem dan memberikan produk free bugs kepada pelanggan.



Automation Testing



Menggunakan Alat Otomatisasi untuk Mengeksekusi rangkaian kasus pengujian



Manual Testing



Manual Testing dari scenario kadang time dan money consuming

Sangat susah untuk test multilingual program secara manual

Automation tidak butuh intervensi manusia

Manual Testing



Automation meningkatkan kecepatan eksekusi tes

Automation membantu meningkatkan Test Coverage

Manual Testing menjadi melelahkan sehingga error menjadi tinggi



Test case apa saja yang di automated?



- High Risk - Business Critical test cases
- Test cases that are repeatedly executed
- Test Cases that are very tedious or difficult to perform manually
- Test Cases which are time-consuming



Workshop Kualitas Perangkat Lunak

Fungsional dan Non-Fungsional

Fungsional Testing



Tipe dari software testing yang berguna untuk validasi sistem software dari spesifikasi ataupun requirement fungsional



Teknik Fungsional testing:

- ✓ **Testing based on Requirements:** Berisi semua spesifikasi fungsional yang membentuk dasar untuk semua pengujian yang akan dilakukan.
- ✓ **Testing based on Business scenarios:** Berisi semua spesifikasi fungsional yang membentuk dasar untuk semua pengujian yang akan dilakukan.

Apa yang dites di Functional Testing



- ✓ **Fungsi utama:** testing dari fungsi utama aplikasi
- ✓ **Usability dasar:** mengecek apakah user dapat melakukan navigasi dengan mudah di layar tanpa ada kesulitan
- ✓ **Kemudahan dalam akses:** mengecek apakah sistem dapat diakses oleh user dengan mudah.
- ✓ **Kondisi error:** menggunakan teknik testing untuk mengecek kondisi error. Mengecek error message yang ditampilkan.

Non-Fungsional Testing



Tipe software testing yang digunakan untuk mengecek non-fungsional aspek (performansi, reliability dll)

Tujuan Non-Fungsional Testing



- Non functional testing dapat meningkatkan usability, efisiensi, dan portability dari produk
- Dapat membantu mengurangi resiko dan biaya yang berhubungan dengan non-functional aspect dari produk
- Optimisasi dari produk yang diinstal, setup, eksekusi, dan di monitor.
- Mengumpulkan dan memproduksi metric dan pengukuran untuk RnD internal.
- Meningkatkan dan menambahkan pengetahuan dari product behavior dan teknologi yang digunakan.



Karakter dalam non-functional testing



- ✓ Non-functional testing dapat diukur, sehingga tidak ada tempat untuk karakter subjektif seperti bagus, lebih baik, terbaik.
- ✓ Angka paling tepat tidak seperti awal dari proses requirement
- ✓ Penting untuk melakukan skala prioritas dari requirement
- ✓ Meyakinkan bahwa kualitas atribut yang digunakan sesuai dengan kebutuhan.



Security

Availability

Efficiency

Integrity

Reliability

Survivability

Usability

Flexibility

Scalability

Reusability

Interoperability

Portability





Types of Testing

Functional

- Unit Testing
- Integration Testing
- Smoke / Sanity
- User Acceptance
- Localization
- Globalization
- Interoperability
- So on ...

Non-Functional

- Performance
- Endurance
- Load
- Volume
- Scalability
- Usability
- So on ...

Maintenance

- Regression
- Maintenance



Software Testing Types

Functional Testing

Non Functional Testing

Unit Testing

- White Box Testing
- Gorilla Testing

Integration Testing

- Gray Box Testing

System Testing

- End to End Testing
- Black Box Testing
- Smoke Testing
- Sanity Testing
- Happy path Testing
- Monkey Testing

Acceptation Testing

- Alpha Testing
- Beta Testing
- OAT

Security Testing

- Penetration Testing

Performance Testing

- Load testing
- Stress Testing
- Soak Testing
- Volume Testing
- Endurance Testing
- Scalability Testing

Usability Testing

- Exploratory Testing
- Cross browser Testing
- Accessibility Testing

Compatibility Testing



THANKS

Any Question?