



# Software Testing & Quality Assurance

Alif Rezky Maulana  
H071221014



# White-Box Testing

Buku The Art of Software Testing mendalami konsep white-box testing namun dengan pendekatan yang lebih mendasar dan teknis. Myers mengarahkan pembaca untuk memahami bahwa tujuan utama white-box testing adalah mengidentifikasi dan menangkap kesalahan pada tingkat yang sangat rendah, termasuk kesalahan logika, cabang, dan jalur eksekusi.

## Langkah / Proses

1. Memahami Kode yang Akan Diuji
2. Membuat Kasus Uji Berdasarkan Struktur Kode
3. Melakukan Eksekusi Kasus Uji
4. Analisis Hasil
5. Perbaikan dan Pengulangan
6. Melakukan Laporan Pengujian

## Contoh Kasus

Misalkan ada fungsi di aplikasi kalkulator yang menghitung nilai faktorial suatu bilangan. Dalam white-box testing, kita akan melihat bagaimana loop yang menghitung faktorial itu bekerja, kemudian membuat kasus uji untuk setiap kemungkinan (bilangan kecil, besar, nol, bilangan negatif) dan memastikan semua cabang kode diuji.

# Black-Box Testing

Inti dari black-box testing adalah menguji perangkat lunak berdasarkan spesifikasi tanpa melihat kode internalnya. Penguji fokus pada fungsi, antarmuka, dan respons perangkat lunak, serta memastikan hasil akhir sesuai dengan yang diharapkan pengguna.

## Langkah / Proses

1. Menentukan spesifikasi fungsional dari perangkat lunak.
2. Membuat test case berdasarkan persyaratan fungsional.
3. Menjalankan pengujian dengan memberikan input dan memverifikasi output yang dihasilkan.
4. Mengevaluasi apakah output sesuai dengan spesifikasi.

## Contoh Kasus

Pengujian aplikasi login. Penguji memasukkan kombinasi input (username dan password), kemudian memeriksa apakah sistem memberikan respon yang sesuai (akses diterima atau ditolak) tanpa memeriksa bagaimana sistem melakukan validasi di belakang layar.

# Unit Testing

Unit testing didefinisikan sebagai proses pengujian yang berfokus pada pengujian unit terkecil dari perangkat lunak, seperti fungsi, metode, atau kelas, secara terpisah untuk memastikan bahwa mereka berfungsi sesuai dengan spesifikasi yang telah ditetapkan.

## Langkah / Proses

1. Memilih unit (fungsi atau metode) yang akan diuji.
2. Membuat test case yang mencakup semua skenario yang mungkin untuk unit tersebut.
3. Menjalankan pengujian dan mengevaluasi hasilnya.
4. Memperbaiki kode jika hasil pengujian tidak sesuai dengan harapan.

## Contoh Kasus

Misalnya ada fungsi yang mengalikan dua angka. Unit testing akan menguji fungsi ini dengan berbagai kombinasi input, seperti (2, 3), (0, 5), atau (-1, -1), dan memastikan outputnya benar.

# Integration Testing

Dalam Software Engineering: A Practitioner's Approach oleh Roger S. Pressmani, integration testing didefinisikan sebagai proses menguji interaksi antar modul atau komponen yang telah diuji secara terpisah (unit testing) untuk memastikan bahwa mereka bekerja bersama dengan baik.

## Langkah / Proses

1. Mengidentifikasi unit atau modul yang perlu diintegrasikan.
2. Membuat test case untuk menguji interaksi antar-modul.
3. Menjalankan pengujian untuk memverifikasi bahwa modul-modul tersebut dapat berfungsi dengan baik ketika digabungkan.
4. Mengevaluasi dan memperbaiki bug yang muncul.

## Contoh Kasus

Misalkan ada modul "input user" dan modul "validasi login". Integration testing akan memastikan bahwa input dari pengguna dapat diteruskan dengan benar ke modul validasi, dan hasil validasi dikembalikan dengan benar ke antarmuka pengguna.

# System Testing

Pengujian menyeluruh dari keseluruhan sistem perangkat lunak untuk memastikan bahwa aplikasi berfungsi dengan baik sesuai dengan spesifikasi yang ditetapkan.

## Langkah / Proses

1. Menyiapkan lingkungan pengujian yang sesuai.
2. Membuat test case untuk menguji seluruh fungsi sistem.
3. Menjalankan pengujian dengan melibatkan semua komponen dan modul.
4. Mengevaluasi hasil pengujian dan memastikan sistem berfungsi dengan benar.

## Contoh Kasus

Untuk aplikasi e-commerce, system testing akan mencakup pengujian seluruh proses, mulai dari pencarian produk, pemilihan produk, pembayaran, hingga konfirmasi pemesanan, untuk memastikan semua bagian aplikasi berfungsi sesuai spesifikasi.

# Regression Testing

Metode pengujian yang dilakukan untuk memastikan bahwa perubahan atau pembaruan pada kode (misalnya perbaikan bug) tidak mempengaruhi fungsi lain yang sudah ada.

## Langkah / Proses

1. Menentukan bagian-bagian dari perangkat lunak yang terkena dampak perubahan.
2. Menjalankan kembali test case lama untuk memastikan tidak ada regresi (kerusakan fungsi).
3. Menambahkan test case baru jika ada fitur baru atau perubahan signifikan.
4. Mengevaluasi apakah fungsi sistem tetap konsisten setelah perubahan.

## Contoh Kasus

Jika ada pembaruan pada fitur "checkout" di aplikasi belanja, regression testing akan melibatkan pengujian kembali seluruh proses belanja dan fitur terkait lainnya seperti pembayaran, untuk memastikan bahwa fitur-fitur tersebut masih berfungsi dengan baik setelah perubahan.





# Evaluasi Hasil Testing dan Identifikasi Penyebab Masalah

**Contoh Kasus : Pengujian fitur Login pada aplikasi WEB**

Ketika pengguna memasukkan username yang lebih dari 50 karakter, aplikasi mengalami **Internal Server Error** yang seharusnya menampilkan pesan error bahwa “Username terlalu panjang”



# Teknik 5 Why's

**Mengapa aplikasi mengalami "Internal Server Error" ketika username lebih dari 50 karakter dimasukkan?**

- Karena server tidak dapat menangani input username yang melebihi batas panjang yang diizinkan oleh database.

**Mengapa server tidak dapat memproses input username yang lebih panjang dari 50 karakter?**

- Karena tidak ada mekanisme validasi di backend untuk menangani panjang username sebelum mengirimkannya ke database.

**Mengapa backend tidak memvalidasi panjang username?**

- Karena proses validasi panjang username tidak dilakukan di sisi frontend, sehingga input yang melebihi batas panjang dikirim langsung ke backend.

**Mengapa frontend tidak memvalidasi panjang username?**

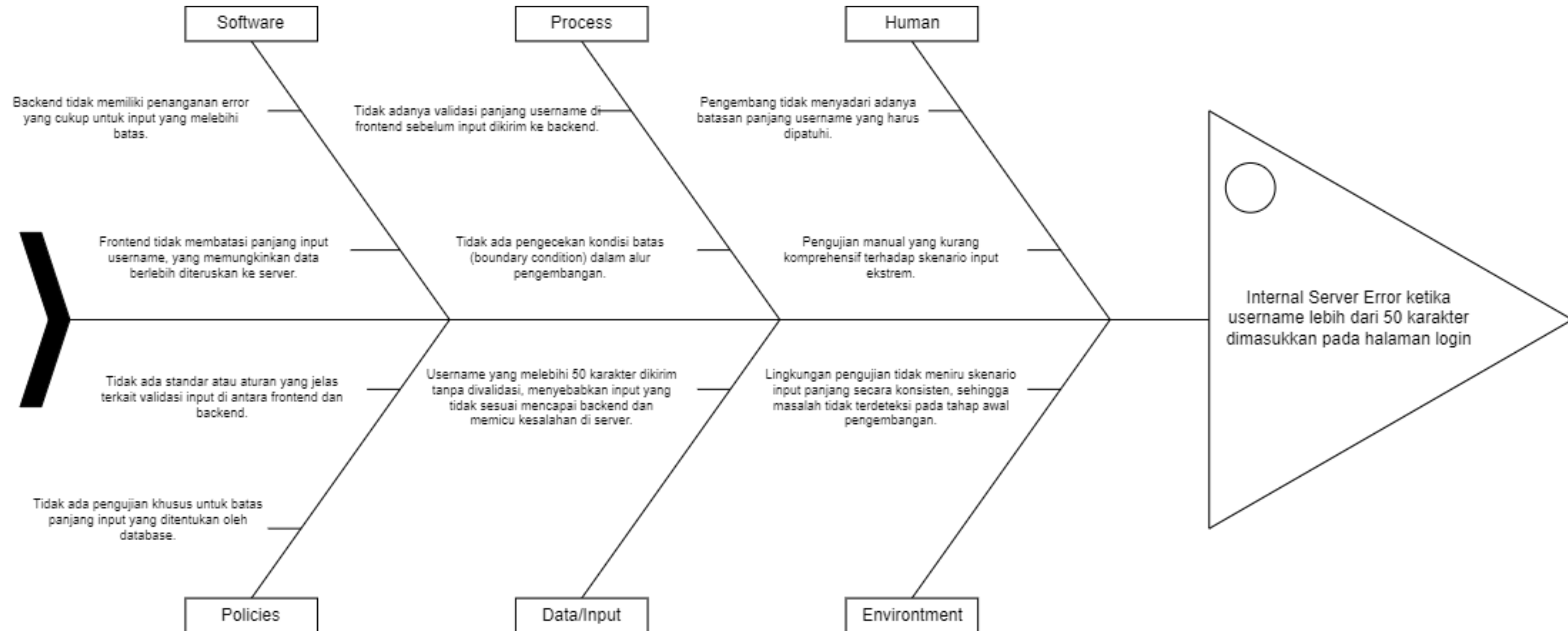
- Karena pengembang belum menambahkan batasan panjang input username di frontend sesuai dengan batas yang diterapkan di database

**Mengapa pengembang tidak menambahkan batasan panjang input di frontend?**

- Karena saat perancangan validasi, pengembang mungkin tidak menyadari atau lupa bahwa ada batasan panjang di database yang harus dijaga oleh frontend juga.

## FISHBONE

# Fishbone



# Thank You