

## Pertemuan 13

# PENGUJIAN APLIKASI WEB

# 1. PENGERTIAN

- Adalah serangkaian aktivitas yang tujuannya untuk menemukan kesalahan dalam isi, fungsi, kegunaan, kemampuan navigasi, kinerja, kapasitas dan keamanan aplikasi web sebelum aplikasi-aplikasi web yang dibuat dikirimkan ke *end user*.
- Hal ini penting karena jika *end user* menemukan kesalahan yang membuat mereka meragukan aplikasi web tersebut, mereka akan pergi ke web lain untuk mencari isi dan informasi
- Langkahnya dimulai dengan fokus pada aspek aplikasi web yang terlihat oleh *user* dan berlanjut pada pengujian yang terkait dengan teknologi dan infrastruktur.

## 2. KONSEP PENGUJIAN UNTUK APLIKASI WEB

### A. Dimensi Kualitas

Kualitas dievaluasi dengan menerapkan serangkaian tinjauan teknis yang melihat berbagai elemen dari model perancangan dan dengan menerapkan proses pengujian.

# Atribut Dimensi Kualitas

## 1. Isi (*content*)

Dievaluasi di tingkat sintak dan semantik. Pada tingkat sintak dokumen berbasis teks diuji dalam hal ejaan, tanda baca dan tata bahasa. Pada tingkat semantik aspek yang dinilai adalah kebenaran informasi yang disajikan, konsistensi di seluruh objek isi dan objek terkait, dan rendahnya ambiguitas

## 2. Fungsi

Diuji untuk menemukan kesalahan yang menunjukkan ketidaksesuaian dengan persyaratan *customer*

## 3. Struktur

Dinilai untuk memastikan bahwa aplikasi web benar-benar menyediakan isi dan fungsi aplikasi web

# Atribut Dimensi Kualitas (Lanjutan)

## 4. Kegunaan

Diuji untuk memastikan bahwa setiap kategori *user* didukung oleh antarmuka yang *user friendly* serta menerapkan semua sintak dan semantik navigasi yang diperlukan

## 5. Kemampuan untuk dapat dinavigasi

Diuji untuk memastikan bahwa semua sintak dan semantik navigasi dilakukan untuk menemukan kesalahan, seperti *link* yang salah dan *dead link*

## 6. Kinerja

Diuji di bawah berbagai kondisi operasi, konfigurasi, dan *loading*

## Atribut Dimensi Kualitas (Lanjutan)

### 7. Kompatibilitas

- Diuji dengan menjalankan aplikasi web dalam berbagai konfigurasi *host* yang berbeda baik di sisi *server* maupun *client*

### 8. *Interoperabilitas*

- Diuji untuk memastikan bahwa aplikasi web berantarmuka dengan benar dengan aplikasi lain dan *database*

### 9. Keamanan

- Diuji dengan menilai kerentanan potensial

## Kesalahan Atribut pada Pengujian Aplikasi Web

- a. Pengujian web mengungkap masalah yang didapatkan pertama kali di sisi *client*, melalui antarmuka yang implementasinya pada sebuah *browser* atau perangkat komunikasi pribadi
- b. Aplikasi web diimplementasikan pada beberapa konfigurasi dan lingkungan yang berbeda, kemungkinan sulit untuk menemukan kesalahan di luar lingkungan tempat kesalahan pertama kali ditemukan

## Kesalahan Atribut pada Pengujian Aplikasi Web

- c. Kesalahan akibat kode program yang tidak tepat (misal HTML), seperti kesalahan penelusuran dokumen
- d. Kesalahan pada *client-server* sulit dilacak di tiga lapisan:  
*client*, *server*, atau jaringan
- e. Beberapa kesalahan yang berada di lingkungan operasi yang bersifat statis, sementara yang lain terkait dengan lingkungan operasi yang bersifat dinamis



## B. Strategi Pengujian

### Langkah-langkahnya:

1. Model konten untuk aplikasi web ditinjau untuk menemukan kesalahan
2. Model antarmuka ditinjau untuk memastikan bahwa semua *use-case* dapat diakomodasi
3. Model perancangan ditinjau untuk mengungkap kesalahan navigasi
4. Antarmuka pengguna diuji untuk mengungkap kesalahan dalam presentasi dan/atau mekanik navigasi
5. Komponen fungsional diuji untuk setiap unit

## Langkah-Langkah Strategi Pengujian (Lanjutan)

6. Navigasi untuk seluruh arsitektur diuji
7. Aplikasi web diimplementasikan dalam berbagai konfigurasi lingkungan yang berbeda dan diuji kompatibilitasnya
8. Pengujian keamanan dilakukan dalam upaya untuk mengungkap kelemahan aplikasi web
9. Pengujian kinerja dilakukan
10. Aplikasi web diuji oleh *end user*, hasil interaksinya dievaluasi untuk menemukan kesalahan isi dan navigasi, keamanan, keandalan, dan kinerja aplikasi web

## C. Perencanaan Pengujian

Sebuah rencana aplikasi web mengidentifikasikan:

- a. Himpunan tugas-tugas yang diterapkan ketika pengujian dimulai
- b. Produk kerja yang dihasilkan ketika setiap tugas pengujian dijalankan
- c. Cara dimana hasil pengujian dievaluasi, dicatat, dan digunakan kembali saat pengujian regresi dilakukan

## 3. PENGUJIAN ISI

Pengujian isi menggabungkan baik peninjauan maupun pembuatan *test case* yang dapat dilaksanakan

### A. Tujuan Pengujian Isi

- Untuk mengungkap kesalahan sintak dengan memeriksa ejaan dan tata bahasa otomatis
- Untuk mengungkap kesalahan semantik yang fokus pada informasi pada setiap isi objek
- Untuk mencari kesalahan dalam pengaturan atau struktur isi dalam susunan dan hubungan yang tepat

Contoh:

KATEGORI
BAJU (4)
CELANA (3)
DOMPET (2)
ROMPI (2)
SWEETER (1)
TAS (4)
TOPI (3)

Penulisan sintak yang salah, dapat memberikan semantik yang berbeda



## B. Pengujian Basis Data

Pengujian basis data menjadi sulit dikarenakan:

- a. Permintaan dari *client* jarang disajikan dalam bentuk yang dapat dimasukkan ke sistem manajemen basis data
- b. Basis data dapat berada jauh dari *server*
- c. Data mentah yang diperoleh dari basis data harus dikirim ke *server* aplikasi web dan diformat dengan benar untuk pengiriman selanjutnya kepada *client*
- d. Objek isi yang bersifat dinamis harus dikirim ke *client* dalam bentuk yang dapat ditampilkan kepada *end user*

## Pengujian Basis Data (Lanjutan)

Pengujian basis data harus memastikan bahwa:

- a. Informasi yang valid dilewatkan antara *client* dan *server* dari lapisan antarmuka
- b. Proses aplikasi web menuliskan ekstraksi atau format data *user* dengan baik dan benar
- c. Data *user* diberikan dengan tepat untuk fungsi transformasi data pada sisi *server* yang membentuk *query* yang sesuai
- d. Query yang dilewatkan ke *layer* manajemen data yang berkomunikasi dengan rutin-rutin akses basis data terletak di komputer lain

## 4. PENGUJIAN ANTARMUKA

- Untuk memastikan bahwa *script* yang benar dibangun untuk setiap permintaan *user* dan benar dikirimkan ke *server*.
- Verifikasi dan validasi antarmuka *user* terjadi pada:
  - a. Model antarmuka memastikan bahwa telah sesuai dengan kebutuhan *stakeholder* dan elemen lain.
  - b. Model perancangan antarmuka ditinjau untuk memastikan bahwa kriteria kualitas generik telah ditetapkan untuk semua antarmuka.
  - c. Selama pengujian fokus pada interaksi *user*.

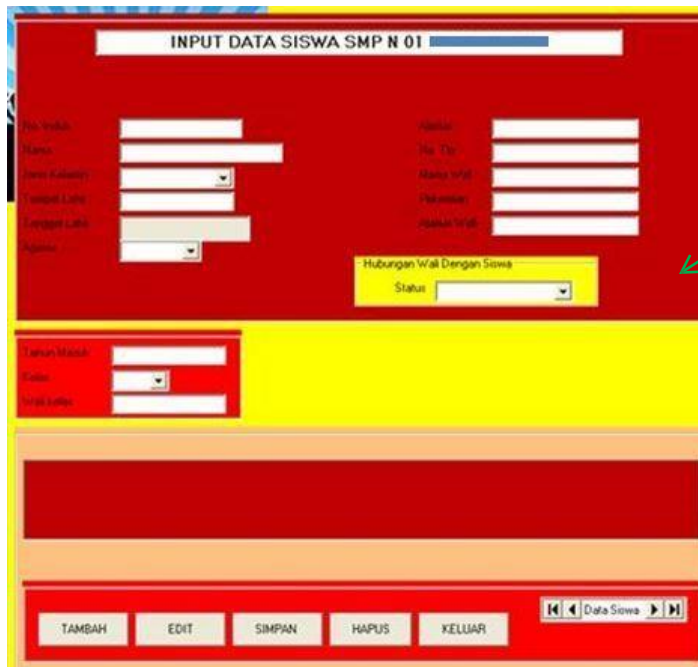


# A. Strategi Pengujian Antarmuka

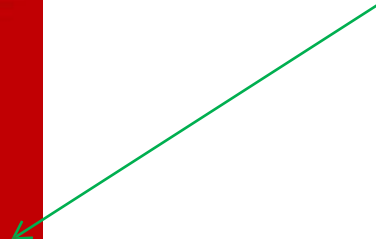
Langkah-langkahnya:

1. Fitur-fitur antarmuka diuji seperti jenis huruf, warna, gambar, *border*, tabel dll
2. Mekanisme antarmuka diuji dengan cara yang sama dengan pengujian unit, misalnya pengujian untuk keranjang belanja pada *e-commerce*, isi *streaming*, penulisan *script* dll
3. Mekanisme antarmuka diuji dalam konteks penggunaan *use case* untuk kategori *user* tertentu
4. Antarmuka lengkap diuji terhadap *test case* terpilih
5. Antarmuka diuji dalam berbagai lingkungan

# Contoh penggunaan warna, teks dan gambar pada antarmuka



Teks sulit dibaca



# Contoh pengujian pada antarmuka yang berbeda

**Menu**

- Home
- Member
- Transaksi
- Cek Buku

**Form Transaksi Peminjaman**

Time : 28 Juni 2013

Id Transaksi : 37

Id Member : Robby

Id Buku : Mengingat Segalanya Dengan E  
Rahasia Puasa Rasulullah Saw &

Data tidak sama

**Menu**

- Home
- Member
- Transaksi
- Cek Buku

**Member**

Pencarian : ID Member atau Nama...

No	Id Member	Nama Member	Email	HP	Aksi
1	1306001	Robby			<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
2	1306002	Andre			<input type="button" value="Edit"/> <input type="button" value="Hapus"/>

## Contoh

Antarmuka diuji dalam berbagai lingkungan:

Lingkungan Administrator tetapi  
login untuk User



Gunakan simbol  
icon yang umum  
untuk exit

## B. Mekanisme Pengujian Antarmuka

1. **Formulir.** Pengujian untuk memastikan:
  - a. Label formulir dapat diidentifikasi secara visual
  - b. *Server* menerima semua informasi form
  - c. *Default* yang tepat saat *user* tidak memilih dari menu *pull down* atau dari tombol
  - d. Fungsi-fungsi perambah seperti tanda panah *back* tidak merusak data yang diisikan ke dalam form
  - e. *Script* yang memeriksa kesalahan *input* data
  - f. Lebar kolom dan jenis data yang tepat
  - g. Mencegah *user* memasukkan *string text* lebih panjang dari jumlah max. yang telah ditetapkan
  - h. Menu *pull down* diurutkan dan dapat dipahami *user*
  - i. *Auto-fill* tidak mengarah ke kesalahan *input* data
  - j. *Key tab* memicu perpindahan di antara kolom

## Contoh input data:

NPM

SILAHKAN ISI USER DAN PASSWORD

Status

Password

Sebaiknya user tidak  
diberikan pilihan dengan  
combo box

## Contoh untuk user yang mengisi data sendiri

**Aplikasi Tiket**

Kode Kereta

Nama Kereta

Jurusan

Harga

Jumlah Tiket

Total Bayar

User memasukkan data sendiri, seharusnya tampil pada saat kode kereta dipilih

## Mekanisme Pengujian Antarmuka (Lanjutan)

2. **Link.** Setiap *link* navigasi diuji untuk memastikan bahwa objek isi atau fungsi yang tepat dapat dicapai
3. **Client-side scripting.** Pengujian dilakukan untuk menemukan kesalahan saat *script* dijalankan
4. **HTML dinamis.** Dijalankan untuk memastikan bahwa tampilan dinamis sudah benar
5. **Pop-up windows.** Memastikan bahwa
  - a. *Pop-up* diukur dan diposisikan dengan benar
  - b. *Pop-up* tidak menutupi jendela aplikasi web asli
  - c. Perancangan *pop-up* konsisten dengan perancangan antarmuka
  - d. *Scroll bar* dan mekanisme kontrol lainnya yang ditambahkan ke *pop-up* diletakkan dengan benar

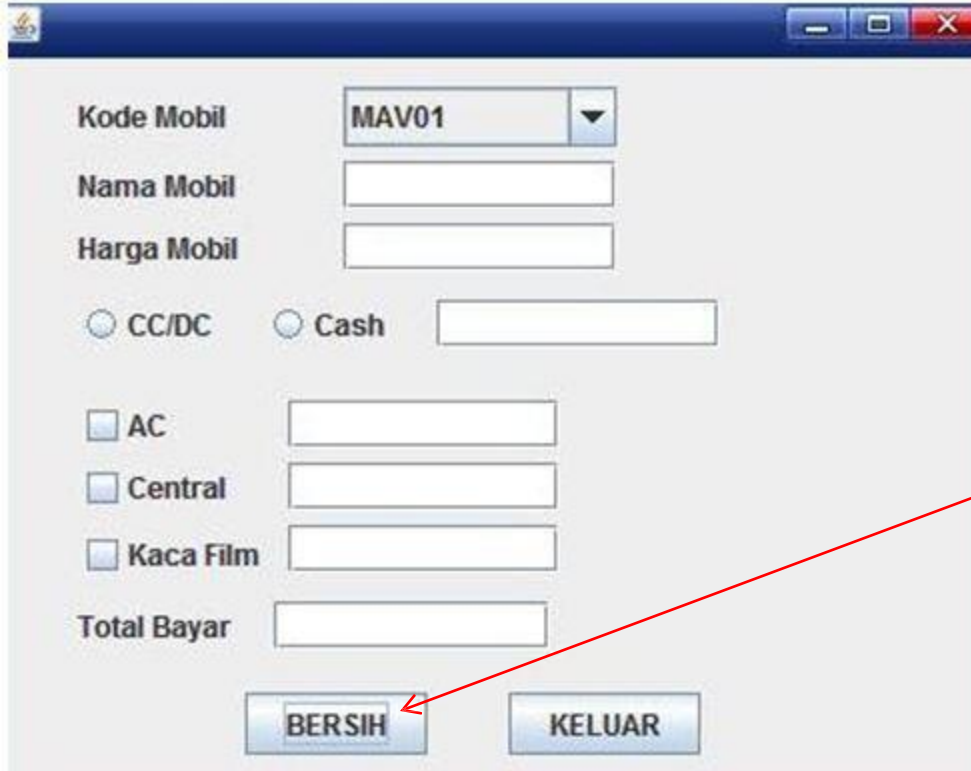


# Mekanisme Pengujian Antarmuka (Lanjutan)

6. **Script CGI.** Pengujian kotak hitam dilakukan dengan penekanan pada integritas data saat dilewatkan ke *script* CGI
7. **Streaming content.** Pengujian menunjukkan bahwa data *streaming* terbaru, ditampilkan dengan benar dan dapat dihentikan tanpa kesalahan dan *restart* tanpa kesulitan
8. **Mekanisme aplikasi antarmuka spesifik.** Pengujian sesuai dengan daftar fungsi dan fitur yang didefinisikan pada antarmuka

# Contoh fitur yang tidak disediakan pada Input Data

## Form Input Data Mobil



Command yang membingungkan dan tidak ada command SIMPAN

## C. Pengujian Kompatibilitas

Aplikasi web harus dapat dijalankan pada komputer yang berbeda, berupa:

- Perangkat tampilan
- Sistem Operasi
- *Browser*
- Kecepatan koneksi jaringan

# Pengujian Kompatibilitas (Lanjutan)

Langkah-langkah uji kompatibilitas:

1. Mendefinisikan sekumpulan konfigurasi komputasi di sisi *client*, mengidentifikasi *platform*, perangkat layar, sistem operasi, *browser* yang tersedia, kecepatan koneksi internet, dll.
2. Melakukan serangkaian uji validasi kompatibilitas berupa pengujian navigasi, pengujian kinerja, dan pengujian keamanan

## 5. PENGUJIAN NAVIGASI

Tugas pengujian navigasi:

1. Memastikan bahwa semua mekanisme yang memungkinkan pengguna aplikasi web melakukan penelusuran melalui aplikasi web.
2. Untuk memvalidasi bahwa setiap unit semantik navigasi dapat dicapai oleh kategori pengguna yang tepat

# Pengujian Sintaks Navigasi

## 1. *Link Navigasi*

Mekanisme menyertakan *link* internal dalam aplikasi web dan *link* eksternal ke aplikasi web lain, dan *anchor* pada halaman web tertentu

## 2. *Redirect*

*Link* beraksi saat *user* meminta URL yang tidak ada atau memilih sebuah *link* yang isinya telah dihapus atau namanya telah berubah

## 3. *Bookmark*

Memastikan bahwa judul halaman yang berarti dapat diekstraksi saat *bookmark* dibuat

# Pengujian Sintaks Navigasi (Lanjutan)

## 4. ***Frame and frameset.***

*Frameset* berisi beberapa *frame* dan memungkinkan untuk menampilkan beberapa halaman web secara bersamaan. Oleh karena itu harus diuji dalam hal isi, tata letak layar dan ukuran yang tepat, kinerja *download*, dan kompatibilitas *browser*.

## 5. ***Site map***

Berisi daftar isi lengkap pada semua halaman web, setiap *entry* harus diuji untuk memastikan bahwa *link* membawa *user* membaca isi/fungsionalitas yang tepat

## 6. ***Search engine internal***

*User* mengetikkan kata kunci untuk menemukan isi yang diperlukan

# 6. PENGUJIAN KONFIGURASI

## A. Masalah di bagian *Server*

- Aplikasi web sepenuhnya kompatibel dengan *server OS*
- Berkas sistem, direktori, dan data yang terkait dibuat dengan benar saat aplikasi web dioperasikan
- Keamanan sistem mengizinkan aplikasi web untuk berjalan dan melayani *user* tanpa gangguan atau penurunan kinerja
- Aplikasi web terintegrasi secara tepat dengan perangkat lunak basis data
- *Script* aplikasi web sisi *server* mengeksekusi dgn benar
- Jika *proxy server* yang digunakan, apakah perbedaan konfigurasi telah diatasi melalui pengujian



## B. Masalah di bagian *Client*

Pengujian konfigurasi fokus pada kompatibilitas aplikasi web pada komponen berikut:

- *Hardware*: CPU, memori, penyimpanan, perangkat cetak
- Sistem operasi
- *Browser*: Firefox, Safari, IE, Opera, Chrome
- Komponen antarmuka: *Active-X, Java applet*
- *Plug in*: Quick Time, RealPlayer
- Konektivitas: kabel, DSL, Wifi

# 7. PENGUJIAN KEAMANAN

Untuk menyelidiki kerentanan lingkungan di sisi *client*, komunikasi jaringan yang terjadi saat data dilewatkan dari *client* ke *server*, dan di lingkungan *server* itu sendiri.

- Contoh kerentanan yang dapat terjadi:
  - *Buffer overflow*, seperti memasukkan URL yang lebih panjang dari ukuran *buffer*
  - Akses tidak sah
  - *Spoofing*
  - Serangan DOS
- Implementasi keamanan:
  - *Firewall*
  - *Authentication*
  - *Encryption*
  - *Authorization*

# Tugas

- Mahasiswa menganalisa web yang sudah ada, kemudian berikan analisis berdasarkan:
  - a. Atribut Dimensi Kualitas (dari atribut ke 1 s/d 9)
  - b. Pengujian Isi, basis data, dan antar muka
  - c. Pengujian kompatibilitas (jika ada)
  - d. Pengujian Navigasi (jika ada)
  - e. Pengujian Keamanan
- Analisis dapat bernilai positif atau negatif
- Tugas dikumpulkan dalam bentuk makalah