



UNIVERSITAS IPWIJA

# ***Aplikasi Web***



# ***Aplikasi Web ?***

**Aplikasi web** adalah suatu program yang memanfaatkan web browser dan teknologi web untuk melakukan sebuah tugas melalui internet.

**Website** adalah sebuah halaman atau sekumpulan halaman web yang saling terhubung dan dapat diakses dari seluruh dunia, selama terkoneksi ke jaringan internet.

# ***Manfaat Aplikasi Web ?***

- Kemudahan akses
- Bebas platform
- Hemat Storage
- Hemat pengeluaran
- Lebih mudah di-monetisasi

# ***Manfaat Aplikasi Web ?***

## **Kemudahan akses**

Kelebihan aplikasi web yang pertama adalah kemudahan akses.

Dengan aplikasi web, pengguna dapat mengakses aplikasi dari mana saja.

Kemudahan ini juga berujung pada jumlah pengguna yang lebih banyak.

User tidak perlu meng-install aplikasi ke PC atau smartphone.

# ***Manfaat Aplikasi Web ?***

## **Bebas platform**

Bebas platform di sini berarti penggunaan aplikasi web tidak terbatas pada satu atau dua sistem operasi saja.

Semua device dan semua web browser bisa digunakan untuk mengakses aplikasi web.

## **Hemat storage**

Dari sisi pengguna, salah satu aplikasi web yang paling terasa jelas adalah hemat storage.

Biasanya, user akan khawatir aplikasi yang mereka download bisa memenuhi memori PC atau smartphone-nya.



# ***Manfaat Aplikasi Web ?***

## **Hemat pengeluaran**

Aplikasi web relatif membutuhkan lebih sedikit pengeluaran daripada aplikasi mobile.

Karena aplikasi web lebih mudah dibuat, di-update, dan dilakukan pemeliharaan (maintenance).

## **Lebih mudah di-monetisasi**

Karena dijalankan pada web browser, user bisa lebih leluasa dalam menggunakan google AdSense atau jenis ads lainnya.

*\* Monetisasi adalah proses mengubah sesuatu agar bisa menjadi penghasilan.*

# ***Ciri dan Sifat Aplikasi Web***

- *Network intensive*
- *Content-Driven*
- *Continuous evolution*
- *Document-oriented*

# ***Ciri dan Sifat Aplikasi Web***

- ***Network intensive***

Sifat dasar dari WebApp (aplikasi web) adalah aplikasi ini ditujukan untuk berada di jaringan dan memenuhi kebutuhan komunitas yang berbeda.

- ***Content-Driven***

Sebagian besar fungsi dari WebApp adalah untuk menyajikan informasi dalam bentuk teks, grafik, audio dan video ke end user.



# ***Ciri dan Sifat Aplikasi Web***

- ***Continuous evolution***

Selalu berkembang secara terus menerus

- ***Document-oriented.***

Halaman-halaman situs yang statis akan tetap ada sekalipun sudah ada pemrograman web dengan java atau yang lain

# ***Karakteristik Aplikasi Web***

## ***# Immediacy***

Diperlukan segera untuk bisa di-publish, dapat dipasarkan dalam waktu singkat.

## ***# Security***

Untuk melindungi isi yang sensitif dan menyediakan pengiriman data yang aman, keamanan suatu WebApp harus diterapkan pada seluruh infrastruktur yang mendukung WebApp dan termasuk dalam WebApp sendiri.

## ***# Aesthetics***

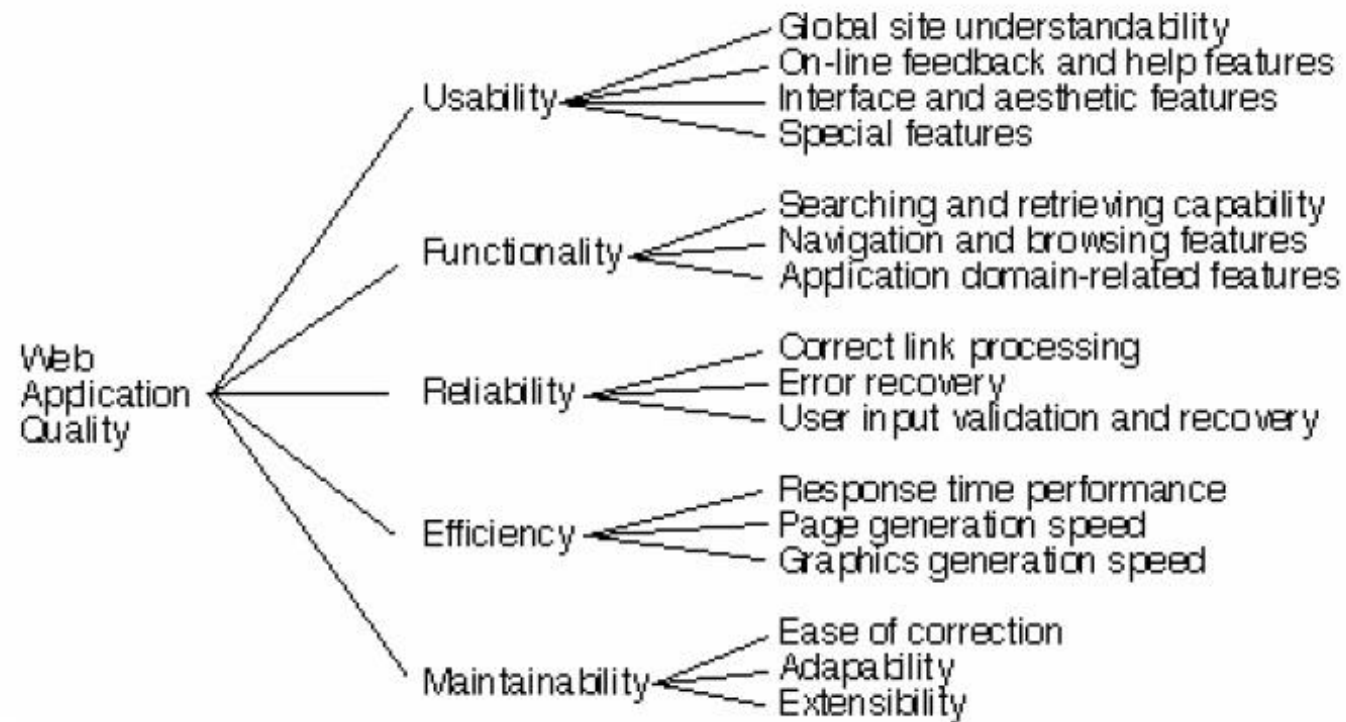
Daya tarik utama WebApp adalah tampilan dan keindahan.

Jika WebApp digunakan untuk memasarkan suatu produk maka sisi estetika harus diperhatikan sebagaimana juga sisi teknisnya.

# ***Fungsi dan Contoh Aplikasi Web***

1. Komunikasi dan Kolaborasi
2. Penyimpanan dan berbagi data
3. *E-commerce*
4. Penyediaan layanan
5. Media dan hiburan
6. Pendidikan dan pembelajaran
7. Penelitian dan pengembangan
8. Layanan kesehatan
9. Administrasi dan manajemen
10. Pembelajaran jarak jauh
11. Dll.

# *Faktor-faktor yang menentukan kualitas suatu web*



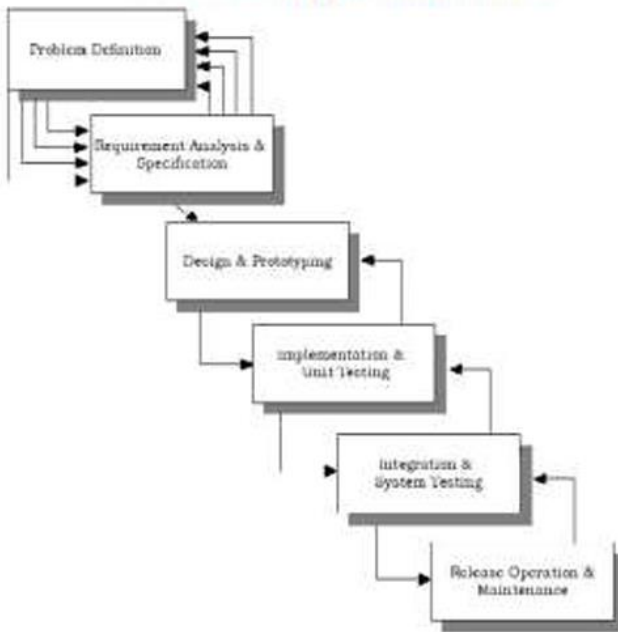
Gambar 1 :Faktor kualitas aplikasi web

# ***Proses Rekayasa Web***

Model yang dianggap cocok dan baik untuk rekayasa web adalah **model modified waterfall dan spiral.**

# Model Modified Waterfall

Modified Waterfall Model

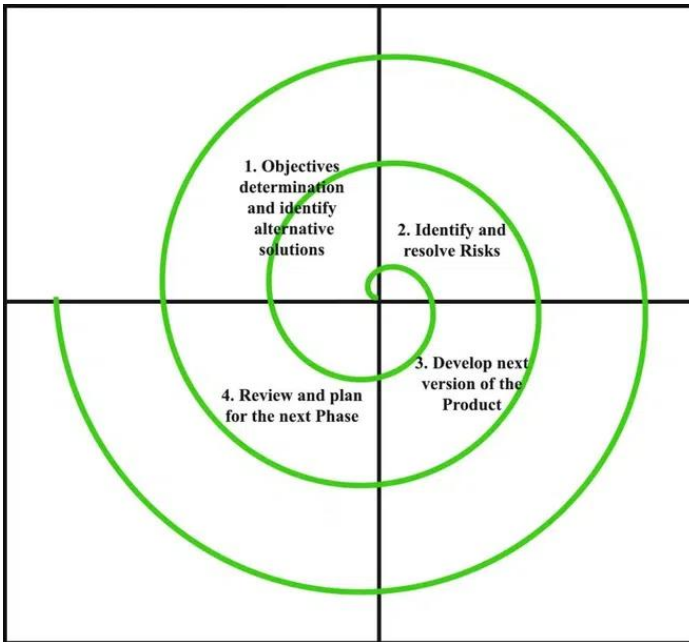


Tahapan dalam modified waterfall adalah:

- **Problem definition and concept exploration**
- **Requirement analysis specification**
- Design prototyping
- Implementation and unit testing
- Integration and system testing
- Operation and maintenance

Pada modified waterfall, perbedaan berada pada 2 proses pertama yang dilakukan secara berulang-ulang sehingga disebut **whirlpool (pusaran)**.

# Model Spiral



Pada model spiral terbagi beberapa sektor

- Determine site objectives and constraints
- Identify and resolve risks
- Develop the deliverables for the interation and verify that they are correct
- Plan the next iteration

# ***Formulasi dan Analisis Sistem Berbasis Web***

Serangkaian aktifitas rekayasa web yang dimulai dengan identifikasi tujuan dan diakhiri dengan pembangunan analisis model atau spesifikasi requirement sistem.



# ***Formulasi***

Formulasi memungkinkan klien dan developer untuk menetapkan tujuan-tujuan pembuatan aplikasi web. Pertanyaan berikut dapat membantu menentukan tujuan:

- **Apa** motivasi utama pembangunan WebApp?
- **Mengapa** WebApp diperlukan?
- **Siapa** yang akan menggunakan WebApp?

# ***Formulasi***

Ada 2 macam tujuan pengembangan aplikasi web:

- **Informational goals**—tujuan dari penyajian isi atau informasi kepada end.
- **Applicative goals**—berkaitan dengan kemampuan yang dimiliki WebApp.

# ***Analisis Rekayasa Web***

Ada 4 tipe analisis dalam rekayasa web:

- ***Content Analysis:*** Isi yang akan disajikan oleh WebApp ditentukan formatnya baik itu berupa text, grafik dan image, video, dan audio.
- ***Interaction Analysis:*** Dijelaskan cara interaksi antara user dan WebApp.
- ***Functional Analysis:*** Menentukan operasi yang akan diaplikasikan pada WebApp dan termasuk di dalamnya fungsi-fungsi yang melakukan proses. Semua operasi dan fungsi dideskripsikan secara detail.
- ***Configuration Analysis:*** Lingkungan dan infrastruktur dimana WebApp akan digambarkan secara detail

\_\_\_\_\_

## Menggambarkan struktur WebApp

- Urutan interaksi sudah bisa dipastikan
- Misal: untuk presentasi tutorial, pemesanan produk yang harus mengikuti urutan tertentu



# Desain Web

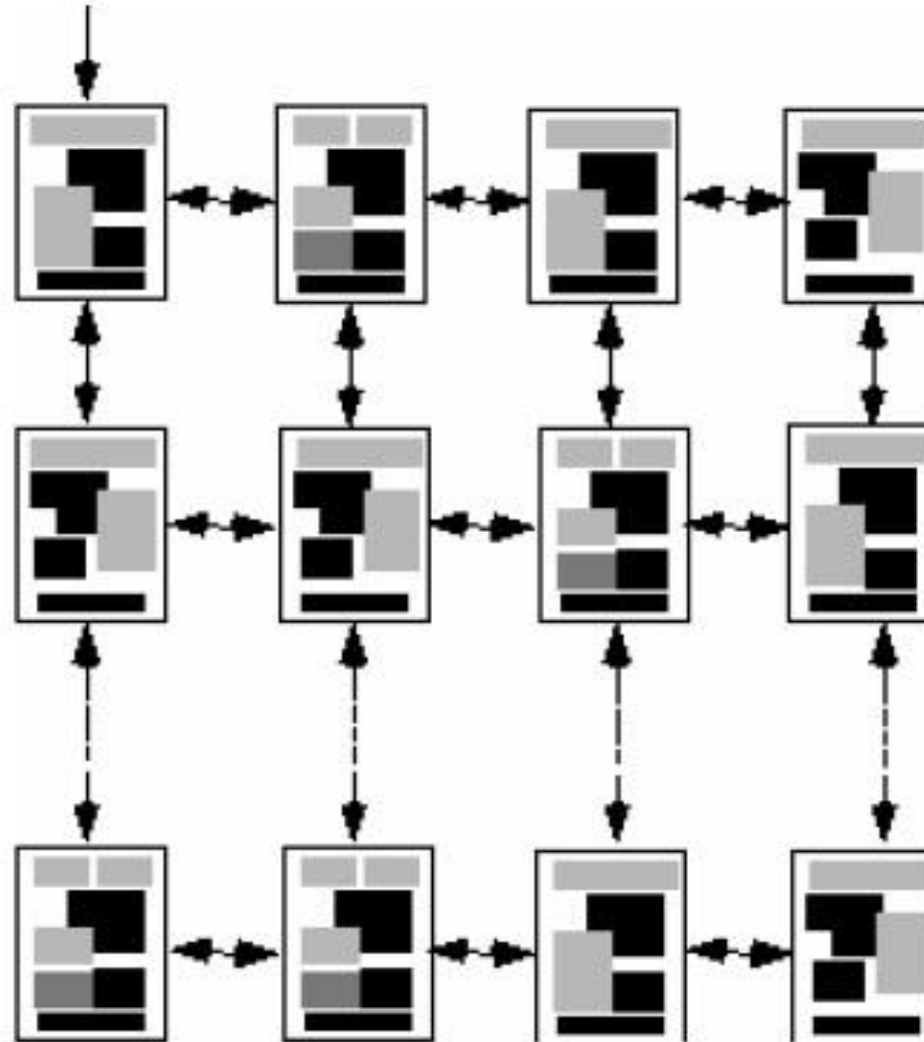
---

## **Architectural design**

Menggambarkan struktur WebApp

### **Struktur Grid:**

- Isi dapat dikategorikan dalam 2 atau lebih dimensi.
- Misal: e-commerce menjual handphone. Horizontal adalah kategori berdasarkan feature hp, sedang vertikal adalah merek HP



Gambar 3: Struktur Grid

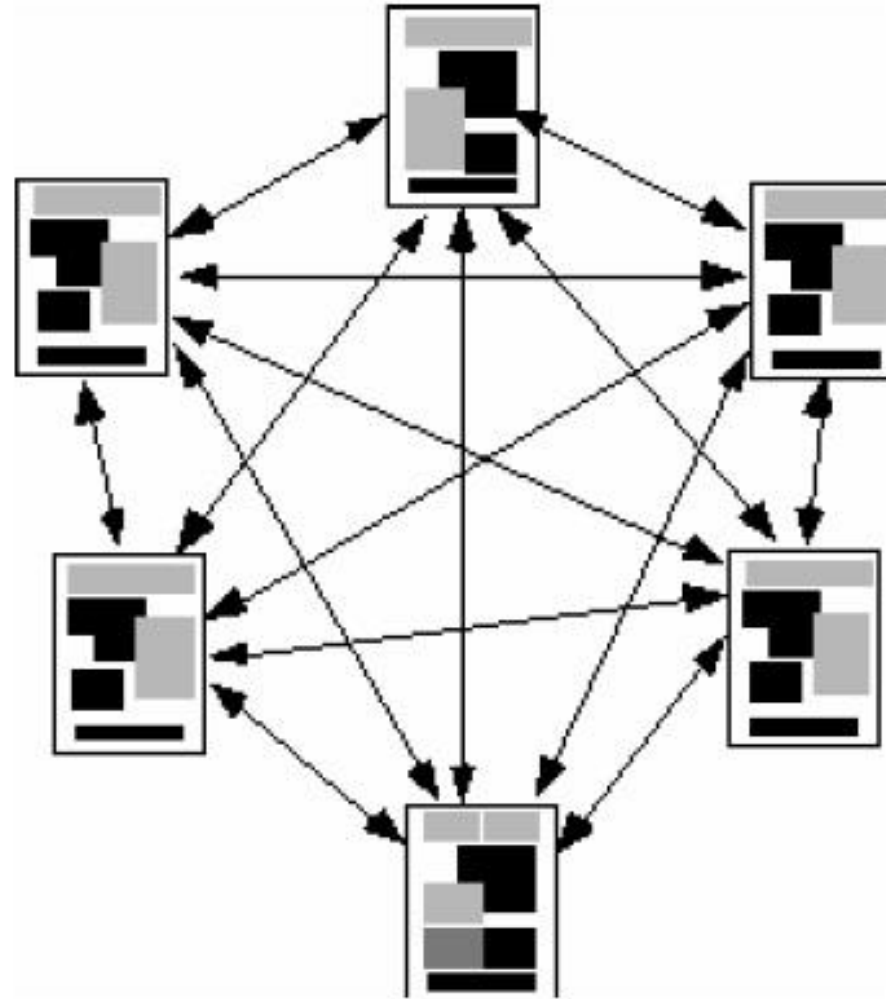
# Desain Web

## **Architectural design**

Menggambarkan struktur WebApp

### **Struktur Jaringan:**

- Komponen pada struktur ini terhubung satu sama lain.
- Sekalipun bersifat fleksibel, struktur ini dapat membingungkan user



Gambar 4: Struktur jaringan atau “pure web”

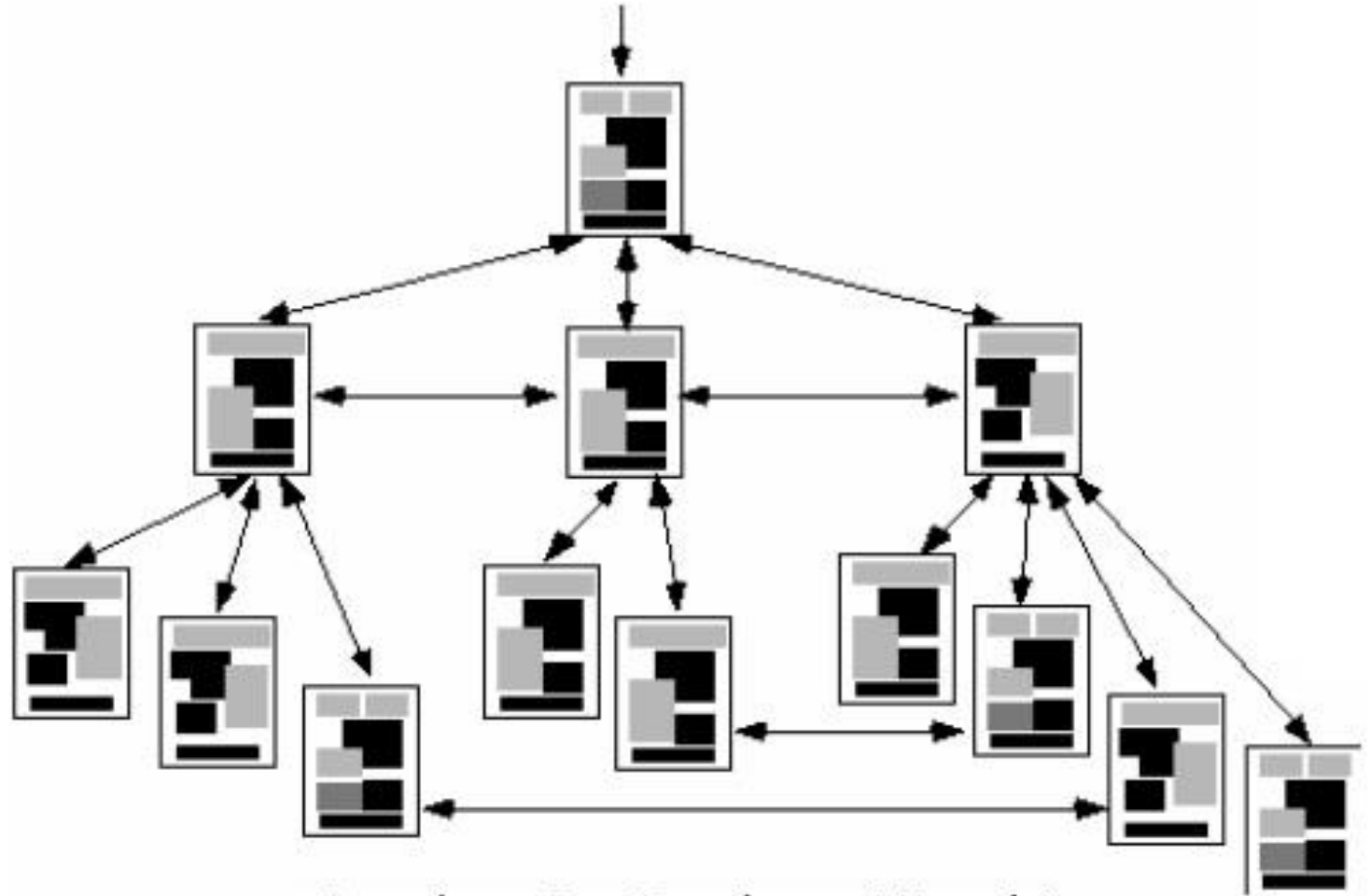
# Desain Web

## **Architectural design**

Menggambarkan struktur WebApp

### **Struktur Hirarki:**

- Struktur paling umum digunakan
- Memungkinkan aliran secara horizontal Selain jalur vertikal yang umum
- Aliran secara horizontal juga bisa mengakibatkan kebingungan user



Gambar 5: Struktur Hirarki

# ***Navigation Design***

Navigation design adalah proses menentukan navigasi halaman-halaman web.

Setelah arsitektur WebApp sudah terbentuk dan komponen-komponen seperti halaman, scripts, applet dan fungsi lain sudah ada, developer menentukan navigasi yang memungkinkan user mengakses isi WebApp dan layanan-layanannya.





# ***Navigation Design***

Pada sebuah aplikasi web, jika user tidak bisa berpindah ke halaman lain dengan mudah dan cepat, maka bisa jadi ini karena grafik dan isi tidak relevant. Ini adalah masalah navigasi.

Dalam desain navigasi beberapa hal perlu dilakukan :

- Menentukan semantik (arti) dari navigasi untuk user yang berbeda.
- Menentukan cara yang tepat: pilihannya adalah text-based links, icons, buttons and switches, and graphical metaphors.



# ***Interface design***

Membangun interaksi dengan user yang konsisten dan efektif.

User interface pada WebApp adalah kesan pertama.

Sekalipun nilai isinya baik, kemampuan prosesnya canggih, layanannya lengkap namun jika user interfacenya buruk maka hal lain tidak berguna, karena akan membuat user berpindah ke web lain.



# ***Interface design***

Pada situs web, user interface design akan berfokus pada pengalaman penggunaan (user experience) dan interaksi.

**Tujuan** dari user interface design adalah untuk membuat interaksi pengguna sesederhana dan seefisien mungkin dalam hal mencapai tujuan pengguna atau yang disering disebut dengan user-centered design.



# ***Fokus Desain Antarmuka***

## **Desain antarmuka inter-modular**

- Dikendalikan oleh aliran data antar modul
- Berikaitan erat dengan desain tingkat komponen

## **Desain antarmuka eksternal**

- Antarmuka antar aplikasi
- Antarmuka antar perangkat lunak dan produsen dan/atau konsumen informasi non-manusia

## **Desain antarmuka manusia-computer**

- Komunikasi antar manusia dan mesin
- Manusia sebagai pengguna memiliki berbagai karakter yang lebih sulit dipelajari.



# ***Three Golden Rules Dalam Desain Antarmuka***

- ***Tempatkan pengguna sebagai pengendali***
- ***Kurangi beban memori pengguna***
- ***Buat antarmuka yang bersifat konsisten***



# **Three Golden Rules Dalam Desain Antarmuka**

## **1. Tempatkan pengguna sebagai pengendali**

- Mendefinisikan interaksi sehingga tidak memaksa pengguna untuk melakukan aksi-aksi yang tidak diperlukannya dan tidak dikehendaki
- Sediakan interaksi yang bersifat fleksibel (pilihan interaksi lain): keyboard, mouse, pena digitizer, layer sentuh / suara
- Memungkinkan interaksi pengguna yang dapat diinterupsi atau dibatalkan
- Sembunyikan hal-hal internal yang bersifat sangat teknis dari pengguna
- Rancangan interaksi langsung dengan objek-objek yang tampak di layer monitor pengguna.



# **Three Golden Rules Dalam Desain Antarmuka**

## **2. Kurangi Beban Memori Pengguna**

- Kurangi short-term memory pengguna (menyediakan ikon visual)
- Memberi default (pilih “reset”)
- Mendefinisikan tombol shortcut yang intuitif. (ALT+P untuk printer)
- Tampilan visual antarmuka pengguna harus didasarkan pada metafor dunia nyata yang familiar (informasi buku bank, layout mesin)
- Tampilan informasi dalam bentuk progresif (informasi umum dilanjutkan dengan detail)

## **Three Golden Rules Dalam Desain Antarmuka**

### **3. Antarmuka yang konsisten**

- Memungkinkan pengguna untuk mengetahui hal yang sedang dilakukan.
- Memelihara sifat konsisten antar sejumlah aplikasi yang serupa.
- Jika model interaktif sebelumnya dibuat berdasarkan harapan pengguna, jangan membuat perubahan apapun tanpa ada alasan yang kuat.





# ***Pengujian pada Rekayasa web***

1. Check isi/informasi untuk kesalahan yang mungkin terjadi, misalnya salah ketik.
2. Design model WebApp di- review untuk menemukan navigation errors.
3. Processing components an Web pages diuji.
4. Integration test untuk arsitektur web
5. Uji WebApp secara keseluruhan setelah disatukan semua komponennya secara lengkap.
6. WebApp yang diimplementasikan pada konfigurasi yang berbeda diuji kompatibilitasnya. Misalnya jika sukses dijalankan di IE, coba di Netscape, dan Firefox.
7. WebApp diuji oleh sekelompok pengguna dengan kemampuan yang berbeda. Bagian yang diuji adalah isi, navigation, kemudahan penggunaan, kehandalan dan unjuk kerja.



# *Selesai*

---

